

Talker Sensitivity To Turn-Taking In Conversation

by

Benjamin Masters

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2024

© Benjamin Masters 2024

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This thesis contains one manuscript in preparation for submission, of which I was the primary author, that was written during the master's program. The manuscript has been directly included as Chapter 2, with minor revisions where necessary to best contribute to the thesis as a whole.

- Masters, B., Aliakbaryhosseinabadi, S., Wendt, D., MacDonald, E. (2024) Pupil response in interactive conversation. *Manuscript in Preparation*

Abstract

Turn-taking in conversation is a complex phenomenon that requires talkers to, at a minimum, simultaneously plan and produce their own speech and listen to and comprehend the speech of their partner(s). Given this necessary division of attention, the increase in listening difficulty introduced by hearing impairments can have confounding effects on a person's ability to communicate, and evaluating listening effort during communication remains difficult. One of the most detrimental effects of hearing loss is the impact it has on one's ability to communicate effectively though, thus the assessment of listening effort in natural environments is especially important.

This thesis takes two approaches to evaluating listening effort in conversation. The first analyzes the response of the pupil at the temporal scale of turn-taking to understand how effort and attention are allocated between speaking, listening, and other task demands. Pupillary temporal response functions to turn-taking are derived and analyzed for systematic differences that exist across people and acoustic environmental conditions, and are further analyzed to determine differences in pupil response based on expected difficulty of a conversation. The second approach analyzes behavioral changes related to the timing of turn-taking to understand how talkers identify that communication difficulty is being experienced by a conversational partner. The floor transfer offset (FTO), defined as the time it takes one talker to begin their turn after another has ended theirs, was manipulated during interactive conversations to mimic the observed increase in magnitude and variability of FTOs in difficult listening environments. To enable this, an audio processing framework was developed to track the state of a conversation in near real-time and manipulate the perceived response time of talkers. The findings suggest that the timing of turn-taking is not used a cue by talkers to infer difficulty.

Acknowledgments

First, I'd like to thank Dr. Ewen MacDonald, my supervisor, for his guidance and mentorship along the way. I'd also like to thank our collaborators at Eriksholm Research Centre, especially Dr. Dorothea Wendt and Dr. Susan Aliakbary Hosseinabadi, for their support on the physiological assessments of conversational effort project, which led to Chapter 2 of this thesis. I am also thankful for the time and effort of my thesis readers, Dr. Evan Risko and Dr. Shi Cao.

I would also be remiss not to express my gratitude to the participants in the studies which make up this thesis, as without them I would have no results to share.

Finally, I thank my friends, family, and loved ones for keeping me grounded along the way.

Table of Contents

Author's Declaration	ii
Statement of Contributions	iii
Abstract	iv
Acknowledgments	v
List of Figures	x
List of Tables	xii
1 Introduction	1
2 Pupil response to turn-taking in interactive conversation	6
2.1 Introduction	6
2.2 Materials and Methods	9
2.2.1 Participants and Experimental Design	9
2.2.2 Speech preprocessing	10
2.2.3 Pupil response preprocessing	10
2.2.4 State change detection in conversation	11
2.2.5 Gaze behavior	12

2.2.6	Estimating pupil responses to state changes	13
2.2.7	Statistical Methods	15
2.3	Results	15
2.3.1	Pupil response across all conversations	15
2.3.2	By condition	16
2.3.3	By turn duration	16
2.3.4	Comparison to grand averaging	17
2.3.5	Analysis of gaze correction	19
2.4	Discussion	21
2.4.1	Interpretation of state change response curves	21
2.4.2	Advantages of the proposed method	24
2.5	Conclusions	25
3	Methods for manipulating conversational dynamics in near real-time	27
3.1	Introduction	27
3.2	Building a processing system	28
3.2.1	Design requirements and constraints	28
3.2.2	Tools	29
3.2.3	User interface	30
3.3	Designing processing modules	30
3.3.1	A callback only example	31
3.3.2	A threaded example	32
3.3.3	Integrating the modules into the framework	33
3.3.4	A more complex example	34
3.4	Performance	37
3.4.1	Round-trip latency	37
3.5	Conclusions	38

4	Sensitivity of talkers to the timing of turn-taking	39
4.1	Introduction	39
4.2	Methods	43
4.2.1	Participants	43
4.2.2	Setup and Equipment	43
4.2.3	Task and Conditions	43
4.2.4	Experimental Procedure	44
4.2.5	The Delay System	45
4.2.6	Data Postprocessing	47
4.2.7	Statistical Methods	48
4.3	Results	48
4.3.1	Delay implementation verification	48
4.3.2	Floor-transfer offsets	48
4.3.3	Interpausal units	51
4.3.4	Pauses	51
4.3.5	Overlaps-within	53
4.3.6	Turn duration and turn-taking rate	55
4.3.7	Speaking and listening time	56
4.3.8	Speech acoustics and articulation rate	57
4.4	Discussion	58
4.4.1	Effects of delay	58
4.4.2	Effects of noise	59
4.5	Conclusions	60
5	Conclusions	61
5.1	Pupil response to turn-taking	62
5.2	Real-time audio processing framework	62
5.3	Delay in conversation	63
5.4	Implications of the findings	63

References	65
Appendix	71
A Supplementary Material	72
A.1 Python class for the implementation and removal of delay in live conversation	72
A.2 Python class for a sub-module that tracks the state of a conversation over time	78

List of Figures

2.1	Four types of conversational state changes	11
2.2	Cumulative distribution functions of state changes relative to each other	14
2.3	State change pupil responses across all conversations	15
2.4	State change pupil responses by condition	16
2.5	Distributions of turn durations	17
2.6	State change pupil responses by turn duration	18
2.7	Longer state change pupil responses by turn duration	18
2.8	State change pupil responses and grand averages	19
2.9	Distributions and measures of gaze behavior	20
2.10	State change pupil responses along with gaze pupil responses across all conversations	20
2.11	State change pupil responses along with gaze pupil responses by condition	21
3.1	Block diagram for the audio processing system	30
3.2	User interface for the audio processing system	31
3.3	Block diagram for a sample static gain module	32
3.4	Python class for a sample static gain module	33
3.5	Block diagram for a sample adaptive gain module	34
3.6	Python class for a sample adaptive gain module	35
3.7	Modifications necessary for implementation of new modules	36
3.8	Delay processing module block diagram	37

4.1	A sample turn-taking exchange	40
4.2	Visualization of the effects of delay implementation	42
4.3	Experimental setup for the delay experiment	44
4.4	Validation of delay implementation	49
4.5	Floor transfer offset results	49
4.6	Interpausal unit results	51
4.7	Pause results	52
4.8	Overlaps-within results	54
4.9	Turn duration and turn-taking rate results	55
4.10	Speaking time results	56
4.11	Speech acoustics and articulation rate results	57

List of Tables

3.1	Round-trip latency results for the audio processing system	38
4.1	Statistical results for the GLMM fit to the floor transfer offset distribution.	50
4.2	Statistical results for the GLMM fit to the IPU duration distributions. . .	52
4.3	Statistical results for the GLMM fit to the durations of pauses.	53
4.4	Statistical results for the GLMM fit to the duration of overlaps-within. . .	54
4.5	Statistical results for the GLMM fit to the distribution of turn durations. .	55
4.6	Statistical results for the GLMM fit to number of syllables per IPU.	58

Chapter 1

Introduction

Communication is a core component of human experience, and conversation is likely the most common manifestation of it. Although day-to-day conversation may often take place without a second thought, it can also be thought of as a complex interactive system that requires participants to simultaneously divide their attention and effort between listening to their partner(s) speech and planning and producing their own speech, among any number of other possible contextual demands.

Perhaps the most consequential effect of hearing loss is the impact it has on communication. Due to the necessary division of effort between speaking and listening, a substantial increase in the amount of effort required for listening can have confounding effects on a person's ability to communicate effectively.

Listening effort has been defined as “the deliberate allocation of mental resources to overcome obstacles in goal pursuit when carrying out a [listening] task” in the Framework for Understanding Effortful Listening (Pichora-Fuller et al., 2016). Evaluating listening effort is an important step in the development of treatment options and diagnostic techniques for the hearing impaired. Typically, listening effort has been assessed using subjective metrics, such as quality assessments, in conjunction with speech intelligibility or speech reception threshold tests (Krueger et al., 2017; Giuliani et al., 2021). However, these methods have faced some criticism for being insufficient for capturing listening effort, as a whole (Winn and Teece, 2021). Giuliani et al. (2021) found physiological metrics, especially pupillometry, to be the most reliable indicators of listening effort.

Understanding listening effort during communication remains difficult given that there are multiple cognitively demanding processes taking place, as opposed to just listening as in speech intelligibility or speech reception threshold tests. Despite this, assessing listening

effort in conversation is an ecologically more valid approach to understanding the impacts of hearing loss on communication. A systematic method for assessing listening effort in conversation would have important implications for the field of audiology, as the ability to use communication as a diagnostic could enable earlier detection of hearing loss, as well as enable detection of so-called "hidden" hearing loss, a phenomena that involves people who exhibit some symptoms thereof, namely difficulty listening in noisy environments, but pass the standard audiometric evaluations (Plack et al., 2014).

Given the complications with evaluating listening effort, specifically, in conversation, the present work instead elects to use the term 'effort' as a more general substitution, which can be interpreted to mean the allocation of mental resources toward the completion of a task. The 'task' in this context can refer to any event or process demanding resources in that moment, such as listening, speaking, performing a visual search, etc. Despite this general substitution, a goal of this work is moving towards the assessment of listening effort in conversation, but at present the methods and interpretation are not concrete enough to use the term 'listening effort' definitively.

Turn-taking can be thought of as a set of transition points in conversation around which a person redistributes their finite attentional resources from primarily speaking to listening, or vice versa. The reallocation of effort between speaking and listening is also likely to change significantly based on the environment that the conversation is taking place in. For example, one may need to focus more effort on listening in conversations at a noisy cocktail party compared to a library, or focus more effort onto deep thought in an intellectually stimulating conversation when compared to small talk.

Sacks et al. (1974) attempted to systematicize turn-taking in conversation, and presented a set of 14 'grossly apparent facts', most of which describe how some aspect of conversation is variable (such as turn duration, turn order, talker count, etc.). Remarkably, despite these considerations, fluid turn-taking can still take place. Levinson and Torreira (2015) propose a psycholinguistic model that describes a simultaneous overlap of speech comprehension, planning, and production, suggesting that speech planning is ongoing throughout a conversational partners turn, and that turn-ending cues are constantly monitored for. This implies that although effort may become primarily directed to either speaking or listening, it is likely that people must still direct attention to planning speech while listening, and predicting the content of their partner's speech while speaking. As mentioned before, this overlap of task demands introduces difficulty in evaluating the impacts of hearing loss on communication, as listening is not the only cognitive process involved.

One possible approach for assessing listening effort in conversation is to monitor physio-

logical signals, such as pupil size. The size of the pupil has long been shown to be indicative of cognitive effort exerted (Kahneman and Beatty, 1966; Beatty, 1982). Of course, analyzing pupil response in conversation is not without its challenges, as the pupil responds to far more than just cognitive effort, including the well-known effect of light exposure, changes in the depth of focus (Kasthurirangan and Glasser, 2005), and even emotional arousal (Bradley et al., 2008). However, if talkers are in-fact reallocating their attention and effort around turn-taking in a temporally consistent manner, then we may be able to detect these changes through analyzing the time course of pupil dilation over many turns and conversations. This approach attempts to infer the cognitive sensitivity of talkers to turn-taking. That is to say, we are interested in understanding how effort and attention, as cognitive resources, are distributed during different phases of a conversation, and how a redistribution of these resources occurs around turn-taking.

Another approach for analyzing listening effort in conversation is to measure the behavioral dynamics of turn-taking, which have been characterized using temporal metrics such as interpausal units (IPUs), which are continuous segments of speech by one talker, floor transfer offsets (FTOs), defined as the time it takes for one talker to begin their turn after (or before) another has ended theirs, and pauses in speech. Turns in conversation are made up of a contiguous set of IPUs and pauses, and separated by FTOs. Previous research has demonstrated that talkers FTOs become longer and exhibit more variability when conversation becomes difficult (e.g., through the addition of background noise, usage of a second language, or hearing impairment (Sørensen et al., 2021; Petersen et al., 2022; Sørensen et al., Submitted)). These observations have led to the general inference that changes in conversational dynamics can be used to infer difficulty experienced by the interlocutors.

An extension of the behavioral analysis approach to listening effort could be to simulate changes in the conversational dynamics without introducing difficulty. One potential implementation of this would be to simulate increases in FTOs by delaying a talker’s speech transmission to their conversational partner. In this case, it could appear to the partner that the talker is experiencing difficulty given their delayed responses, which may exhibit some adaptive behavior from the partner, indicating that talkers use the timing of their partner’s turn-taking to infer level of effort or difficulty. From this behavioral study, we aim to understand how talkers are sensitive to their partner’s turn-taking dynamics, which we infer are representative of effort as a result of conversational difficulty. Its hypothesized that talkers will identify the changes in turn-taking dynamics introduced by the delay, and adapt their behavior in response, suggesting that talkers are sensitive to the timing of turn-taking in conversation.

Some studies have used subjective metrics to assess listening effort, as well, and found

them to be correlated with expected increases in effort (Zekveld et al., 2011; Mackersie et al., 2015). However, one of the goals of this work is a movement towards objective metrics of listening effort, which have been shown to be reliable and sensitive, and tend to translate better across tasks and experiments (Giuliani et al. (2021)). One reason for this shift toward objective measures of effort is that subjective rating scales can typically only be performed on an experimental or, at a minimum, a trial (conversational) level. Therefore, no information can be gained about how effort varies at different points throughout a conversation without causing disruption. Further, it's well understood that listening and conversing becomes more difficult in noisy environments (Beechey et al., 2018), and it's been shown that talkers adapt their behavior to a partner's difficulty (Hazan and Baker, 2011; Beechey et al., 2020). For these reasons, in the following studies it is not of much benefit to evaluate subjective experiences, as for the pupillometry study subjective experiences cannot be evaluated at a fine enough temporal scale, and for the behavioral study, we are probing for a reaction to suspected difficulty rather than evaluating difficulty.

This thesis attempts to systematically analyze the sensitivity of talkers to turn-taking in conversation to infer not only how talkers divide their attention and effort, but also how they perceive their partner's processing demands. Both methods attempt to further our understanding of listening effort in conversation, and work towards the goal of naturalistic and ecologically valid assessments of the effects of hearing loss on communication. The first approach analyzes physiological changes around turn-taking in an attempt to better understand divided attention. The second approach manipulates the timing dynamics of turn-taking to determine if talkers use their partner's response time as a cue to infer they may be experiencing difficulty. Also introduced is a framework for implementing audio processing during interactive conversations in near-real time, enabling a new type of conversational experiment where the turn-taking dynamics are manipulated during the conversation, the first of which is outlined in the second approach.

Thesis Organization

Chapter 2 explores using physiological measures to study effort in conversation. A method is introduced which enables the derivation of pupillary temporal response functions to conversational turn-taking. By comparing pupil responses between conversations with different background conditions (quiet vs. noise), we can infer how effort changes as an effect of conversational difficulty.

Chapter 3 introduces a python-based real-time audio processing framework that enables a new type of experimental design, in which researchers can manipulate the dynamics of a conversation as it's taking place.

Chapter 4 takes advantage of the framework introduced in Chapter 3 to artificially increase the floor transfer offsets perceived by talkers in interactive conversation. This study aims to determine if the timing of a talker's turn-taking is used by conversational partners to infer that they may be experiencing difficulty.

Chapter 5 presents a summary of the work and discusses potential future directions and applications of the findings.

Chapter 2

Pupil response to turn-taking in interactive conversation

2.1 Introduction

In recent decades, pupillometry has become a well-known tool for assessing cognitive effort. The so-called task-evoked pupillary response (TEPR) refers to the dilation of the pupil in response to an increase in mental effort and load induced while performing a variety of cognitively demanding tasks, including memory recall, language processing, and quantitative reasoning (Beatty, 1982; Kahneman and Beatty, 1966).

In addition to explaining general cognitive effort, the TEPR also has specific applications in auditory sciences, where pupil response has been interpreted as an indicator of listening effort. For example, pupil dilation has been observed to reflect increased effort based on the signal-to-noise ratio in speech intelligibility tasks (Wendt et al., 2018; Zekveld et al., 2010) and age and hearing status in speech reception threshold tests (Zekveld et al., 2011). The pupil response also indicates increased effort when switching attention between acoustic sources (McCloy et al., 2017), and when dividing attention between multiple streams of speech simultaneously rather than focusing on one (Koelewijn et al., 2014).

For these studies, it has been standard to report summary metrics of pupil response, such as mean and peak dilation, by evaluating changes in pupil size relative to a baseline window immediately before presentation of a stimulus. However, evaluating effort in dynamic, interactive environments presents challenges. In such contexts, sequential stimuli may have overlapping cognitive effects, making it difficult to clearly define these baseline and response windows.

Interpreting pupil response as a measure of effort is further complicated by the fact that a variety of other factors can induce dilations or constrictions of the pupil, such as emotional arousal (Bradley et al., 2008), depth of focus (or accommodation) (Kasthurirangan and Glasser, 2005), and light exposure. There can also be measurement artifacts due to distortion of the shape of the pupil during blinking and eye movement that can have confounding effects (Gagl et al., 2011; Yoo et al., 2021).

Although listening effort can be directly inferred from pupil response in purely auditory experiments, it is more difficult to evaluate in situations where there are other cognitive demands. For example, when a participant response is required, e.g., in the form a button press, pupil size has been shown to be significantly larger and more sustained than when a response is not required (Privitera et al., 2010). For this reason, pupil dilation during participant response is typically disregarded. In interactive environments, such as conversation, there is consistent overlap from multiple cognitively demanding processes. For example, participants must both listen to and comprehend their partner’s speech and plan and produce their own speech. Pupil response has recently been evaluated in conversation by considering first-order statistics of pupil size during separate phases of the conversation or in different conversational conditions. Li et al. (2020) found that pupil response between speaking and listening times varied significantly during tasks with a lower communication load, but not during tasks with a higher load. Aliakbaryhosseinabadi et al. (2023) observed a larger pupil size when conversing in noise than in quiet. These studies take a conversation level approach to analyzing differences in effort between speaking and listening time or based on background condition.

However, assessing listening versus speaking effort in conversation may not be as simple as analyzing the overall cognitive effort exerted while listening or speaking. Levinson and Torreira (2015) suggest that there must be simultaneous predictive components of both comprehension and production of speech for fluid turn-taking to take place, given that there is considerable latency involved in speech production. Therefore, there is value in determining if we can use pupil response to measure differences in effort at a finer temporal scale, such as around turn-taking in conversation where there is likely to be overlap of listening and speaking effort and a reallocation of attentional resources.

Turn-taking is a coordinated process that requires participants to not only listen to their partner(s) and plan their own speech, but also to monitor a variety of other acoustic, behavioral, and contextual cues to interpret when they should take their turn (Brusco et al., 2020; Gravano and Hirschberg, 2011; Hjalmarsson, 2011). Measures of the temporal dynamics of turn-taking, such as interpausal units (IPU) and floor transfer offsets (FTO), have been shown to significantly vary based on background noise, native versus second language, hearing status, and hearing aid amplification (Petersen et al., 2022; Sørensen

et al., 2021), and it’s been suggested that the overall difficulty level of a conversation can be inferred by characterizing the turn-taking dynamics over the course of that conversation.

Our current understanding of turn-taking suggests that there is both predictive and reactive effort exerted during conversation. However, traditional pupillary analysis methods investigate changes in pupil size relative to discrete events or in clearly defined windows, which may not capture the full scope of cognitive effort involved in conversational turn-taking. For example, if the reference point is defined as when a person starts speaking, this would potentially mis-attribute effort related to speech planning to listening, as this preparation begins before talking starts, and potentially even result in a considerably effortful baseline window. Due to the difficulty of defining clear response windows, we propose that the analysis of the pupillary responses as a time series is likely to have the most informative results for assessing divided attention and effort in conversation.

Some studies have assessed the temporal dynamics of pupil responses. Wierda et al. (2012) used a deconvolution approach to identify differences in pupil responses when people are presented with a single visual stimulus versus multiple sequential stimuli, and found clear peaks in dilation corresponding to the number of stimuli presented. Others have analyzed the time course of pupil response to various stimuli by fitting an Erlang gamma function (Hoeks and Levelt, 1993; McCloy et al., 2017), fitting polynomials of varying orders using growth curve analysis (e.g., 4-th order polynomial models in Wagner et al. (2019)), or through generalized additive mixture models (Van Rij et al., 2019). However, these studies again operate on an experimental design that is expecting a causal response to some event. In the context of conversation, we want to measure predictive components to turn-taking as well and expect that assuming a general shape of the pupillary response function may not be appropriate, as there is not a clear stimulus to measure the response to. It must also be noted that when interpreting the pupil response over time, careful attention must be given to the response latency of the pupil, as it can take a significant amount of time for the pupil to react to presentation of a stimulus (up to 200 ms in response to light and up to 600 ms when shifting focus from far to near) and multiple seconds more for the pupil to reach its peak dilation or constriction Mathôt (2018).

One possible approach to analyzing pupil response in conversation is to model how pupil size changes as a function of turn-taking. The problem of estimating one signal from another exists in many different domains and has well been explored. One solution to this problem is to estimate a temporal response function (TRF), which is a linear filter (or kernel) that is derived to optimally map from one signal to another (Theunissen et al., 2000). It is often used in neuroscience to model how the brain responds (e.g., via EEG or MEG) to a continuous stimuli, such as the acoustic envelope of speech (Ding and Simon, 2012). However, the underlying math is applicable to any set of time series.

Additionally, the model can be generalized to include multivariate input signals, and a temporal response function will simultaneously be derived for each. However, to apply this approach to analyzing pupil response in conversation, there must first be a defined signal to measure the pupil response to. Given that our goal is to estimate how effort changes around turn-taking, we propose defining the input signals as the start and end of turns.

In the present study, we investigate an approach that estimates pupillary temporal response functions to turn starts and ends during conversation. By doing so, we aim to develop a method with the potential to disentangle and investigate cognitive processing related to speaking and listening using pupil response measurements. We also compare the temporal response functions derived for conversations taking place in quiet versus noise, as background noise is well known to significantly impact communication difficulty. We additionally expect that turns of different lengths require different amounts of preparation or listening effort, for example a longer turn should require more time and effort for speech planning from the talker, and a longer period of sustained effortful listening from the listener. Therefore, a comparison of TRFs fitted to data from the starts and stops of short vs. long turns is made.

2.2 Materials and Methods

2.2.1 Participants and Experimental Design

The data analyzed here was previously collected as part of Aliakbaryhosseinabadi et al. (2023). In summary, 12 pairs of older (average age of 63.2 ± 6.4 years) Danish talkers were recruited. The experimental procedure began with participants signing an information and consent letter. Following this, a hearing screening was conducted by a qualified audiologist to verify that all participants met age-adjusted normal-hearing thresholds, as specified in ISO-7029. Participants were seated face-to-face and performed a subset of the DiapixUK spot-the-difference tasks modified to include Danish signage (Baker and Hazan, 2011). Two practice sessions were performed, the first to familiarize participants with the task, and the second to familiarize participants with the measurement equipment used during the study.

Following the practice sessions, conversations took place in a randomized order under the following conditions: Quiet, 60 dBA noise, 70 dBA noise, and simulated conductive hearing loss (SHL). In the SHL condition, the participants wore earplugs that provided, on average, 25 dB of attenuation. In the noise condition, a calibrated loudspeaker array played noise at the appropriate level. Two replicates of each condition were performed,

resulting in 8 total conversations for each pair. Speech signals were recorded using headset microphones. Pupil response and eye gaze behavior were recorded using Tobii Pro 3 eye tracking glasses.

Participants provided informed consent, and the experiment was approved by the Science Ethics Committee for the Capital Region of Denmark (No. H-16036391). Participants were remunerated as a thanks for their participation. Secondary analysis of the data performed at the University of Waterloo was approved by the university’s Research Ethics Committee as application No. 45442.

2.2.2 Speech preprocessing

Voice activity detection (VAD) was performed on the speech signals using individually defined root mean square (RMS) thresholds in 5 ms windows with 1 ms of overlap. Windows containing an RMS power greater than the threshold were classified as containing speech and windows with a power below the threshold as not containing speech. As recommended in Heldner and Eklund (2010), segments containing less than 90 ms of continuous speech were re-assessed as non-speech acoustic bursts, and quiet segments less than 180 ms were interpreted as brief pauses within speech, and both were bridged over. VAD signals were then resampled using nearest-neighbor interpolation to 20 Hz.

2.2.3 Pupil response preprocessing

The pupil diameter data was extracted from the Tobii recordings and time-aligned to the speech signals. Any sample greater than 3 standard deviations from the mean was identified as an artifact due to blinking. A fixed window of 50 ms before and 150 ms after each of these artifacts was removed to mitigate the effects of blinking on pupil response, as suggested by Winn et al. (2018). The eye with the least missing data was selected for further analysis for each person in each conversation. If more than 40% of a participant’s pupil diameter samples in any conversation was missing, their pupil response for that conversation was excluded from further analysis. Based on these criteria, 35 pupil response signals were excluded (18.2%). 31 of the excluded responses belonged to only 4 participants, who may have been particularly susceptible to eye tracking errors. 15 additional responses were excluded due to equipment problems during data collection (7.8%). The distribution of the responses that were removed was relatively balanced across conditions (Quiet: 12, SHL: 17, N60: 11, N70: 10). In the pupil response signals that remained, missing values between

valid samples were interpolated using a cubic spline method. Leading and trailing missing data were set to the mean pupil diameter of that participant for that conversation.

To isolate the pupil response to turn-taking, and to perform a long-term detrending to avoid any initial effects from arousal at the task onset, the signals were band-pass filtered between .1 and 1 Hz, based on previous observations of the turn-taking rate in Diapix conversations generally being between .4-.6 floor-transfers per second (Sørensen et al., 2021). The filtered signals were then down sampled to 20 Hz to reduce computational complexity. The pupil data was then standardized within people, such that the distribution of all pupil diameter measurements across all conditions and replicates for each participant had zero mean and unit variance. This normalization procedure was selected to maintain any changes in the size and variability of the pupil across the different conditions.

2.2.4 State change detection in conversation

Corresponding pairs of VAD signals were input into a conversational state labeling algorithm, which identified the start and end times of turns and IPUs. For the purposes of this method, state changes are defined as the points in time at which speakers start and stop their turns. Given that this experimental setup is dyadic and different responses are expected to occur whether a person is speaking or listening, state changes are further classified into two categories: belonging to oneself or belonging to one’s partner. As seen in Figure 2.1, this classification scheme yields the following four state change events: self-start, self-stop, partner-start, and partner-stop. The state changes are identified as discrete points in time during each conversation and arranged as a set of impulse trains.

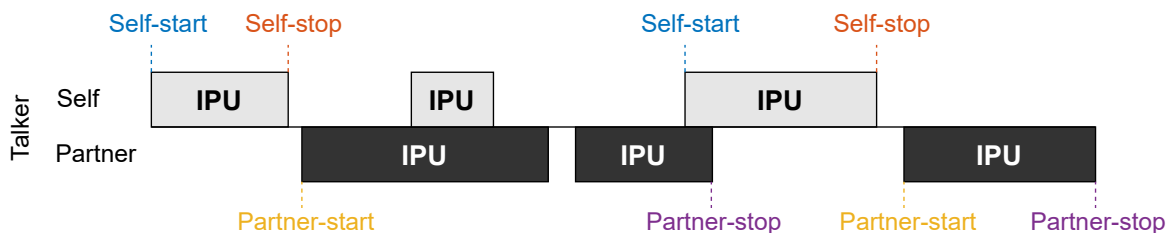


Figure 2.1: The four types of conversational state changes denoted in a sample turn-taking scenario.

2.2.5 Gaze behavior

A measure of gaze distance was used to account for components of the pupil dilation response that could be attributed to the near/far pupil response or differences in luminance between the Diapix image and a person’s conversational partner. One of the eye-related time series provided by the Tobii Pro 3 glasses are the gaze points, which are estimated using the intersection of gaze vectors projected from the center of each eye. The depth component of these points provide an indication of the depth of focus of the wearer. In the experimental setup, participants were seated about 1.5 m apart from each other. Each participant had their Diapix images placed directly in front of them, approximately 40 cm away. Thus, the estimated gaze depth at each point in time can be used to infer whether the participant was looking at their picture or their partner.

The gaze depth estimates were preprocessed by first removing outliers. Rather than using statistical methods to do this, it was instead assumed, based on the experimental setup, that any measurement indicating a gaze point greater than 3m away must be an artifact due to blinking or eye movement. As such, estimates larger than this threshold were removed and replaced by interpolating through the remaining estimates. The gaze depth estimates were then low pass filtered with a cutoff frequency of 10 Hz. To match the sample rate of the VAD and pupil dilation data, the gaze estimates were down sampled to 20 Hz.

To integrate gaze behavior as a covariate into the model, it must be of the same form as the state change signals (one or multiple impulse trains). For this, a threshold gaze depth of 95 cm, halfway between the expected distances of the image and partner, was applied to the Tobii gaze depth estimates. If the depth of a person’s gaze was below this threshold, their fixation target was assumed to be the picture, whereas if it was above the threshold, it was assumed they were looking at their partner. From this, we computed the amount of time in each conversation spent with a participant looking at their partner as the proportion of total samples with a gaze depth estimate greater than the threshold. We also computed the duration of each glance at the partner, as the duration of continuous segments of the gaze depth signals that were greater than the depth threshold. To extract the differences in pupil response that occur at the change between regions, the points in time when a person’s gaze crossed the previously defined threshold were identified. This results in two impulse trains, the first contains the points at which a person looks up at their partner (i.e., transition from near-to-far), and the second contains the points at which they look down at their image (i.e., transition from far-to-near).

One key consideration with this approach is that the signals we are using to determine whether a person is looking at the image or their partner are only based on the depth of a

participant’s gaze, and therefore cannot be definitively stated to belong to these regions. Therefore, careful interpretation of these results must be made, especially given that eye-gaze behavior has been shown to be a significant factor in regulating turn-taking (Degutyte and Astell, 2021). However, the goal of including gaze behavior as a covariate is to capture the pupil response to distance related changes in gaze to avoid misclassifying a near/far pupil response as a response to a conversational state change. Therefore, inclusion of this signal should still capture the pupil response appropriately whether a talker is, in fact, looking at their partner as expected or instead averting their gaze to somewhere else in the room.

2.2.6 Estimating pupil responses to state changes

The initial objective of this analysis is to estimate a general pupillary temporal response function corresponding to each type of conversational state change. To do this, a time-lagged multivariate ridge regression, based on the approach commonly utilized in EEG data analysis (Theunissen et al., 2000), was performed on each conversation. We define the stimulus signals as the set of impulse trains indicating the locations of state changes in a conversation. For the model to account for changes in gaze target (i.e., from the partner to the image, or vice versa), the stimulus also includes the extracted change of gaze target signals. The response signal is the preprocessed pupil data for the same conversation. Thus, the resulting model computes six pupillary TRFs: one for each of the four state changes, one for when a talker looks from the image to their partner, and one for when a talker looks from their partner to the image.

With this modeling approach, there are four hyperparameters to consider: the minimum and maximum time lags, the time lag step, and the regularization parameter. The minimum and maximum time lags dictate the duration of the estimated temporal response function, which can be thought of as a window size. In this analysis, the minimum and maximum lags were selected such that the window spanned 2.5s before and after the state changes. These values were selected based on the expectation that other state changes would likely occur within the window. Figure 2.2 shows that there is an approximately 50% likelihood of each other state change having occurred by 2.5 s in either direction, enabling the model to disentangle the effects from neighboring state changes. The time lag step was selected as 1 sample, at the downsampled 20 Hz rate, to maximize temporal resolution. The regularization parameter was optimized via a 10-fold cross validation performed on each conversation and selected to minimize the average residual sum of squares across all conversations ($\lambda = 4$). This procedure ensures that the same regularization value

is used in the derivation of all models, which prevents artifacts that could be introduced due to differing amounts of smoothing between conversations introduced by the regularization.

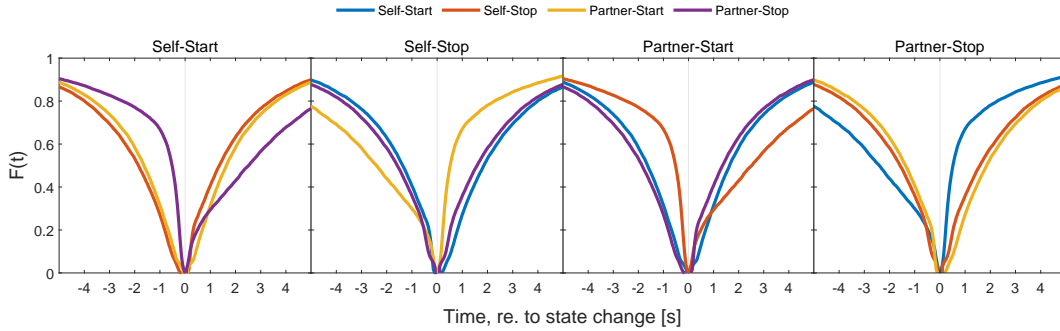


Figure 2.2: Cumulative distribution functions of the state changes relative to other state changes, over time. Each panel corresponds to a different type of state change. The colored curves indicate the probability that each of the other state changes has occurred, as a function of time, with respect to the state change each panel belongs to. In computing these curves, only positive FTOs are included, such that the probability $F(t)$ at time 0 will be 0 for all resulting CDFs.

The temporal response functions to each of the conversational state changes and gaze target changes were estimated for each conversation with the MATLAB mTRF toolbox using the parameters specified above (Crosse et al., 2016). To perform a group level analysis of the results, combinations of these temporal response functions can be averaged together or statistically analyzed as a sample. The set of temporal response functions to be averaged is determined based on the desired analysis. To compare between conditions, for example between quiet and 70 dBA noise, one would average all temporal response functions within those conditions, resulting in a set of pupillary TRFs, with each TRF corresponding to one of the stimulus signals, for each of the conditions.

Given the objectives of this work are to analyze the pupillary response to turn-taking in conversation, the results presented will generally only include the response curves corresponding to the so-called conversational state changes. However, in all results presented the gaze target changes were included as covariates in the model, thus the response curves are derived while accounting for gaze behavior.

2.2.7 Statistical Methods

To assess when the state change responses are significantly different from 0, a pointwise one-sample t-test is performed on the set of time series. To determine if the state change responses are different across groups (such as by condition or turn duration), a pointwise two-sample t-test is performed on the set of temporal response functions belonging to each group. In both cases an alpha value of .01 was used to determine significance.

2.3 Results

2.3.1 Pupil response across all conversations

Figure 2.3 shows the state change responses found by averaging across all conversations, also denoted are the segments within which the curves have a value significantly different from zero ($p < .01$). As an effect of standardization, the amplitude of these curves is proportional to the pupil size, and relative to a talkers average pupil size across all conversations. Significant responses were found to all four state changes. The segments of significance in the results reveal that there is a systematic pupil response around turn-taking in conversation that exists across people, conditions, and turn characteristics.

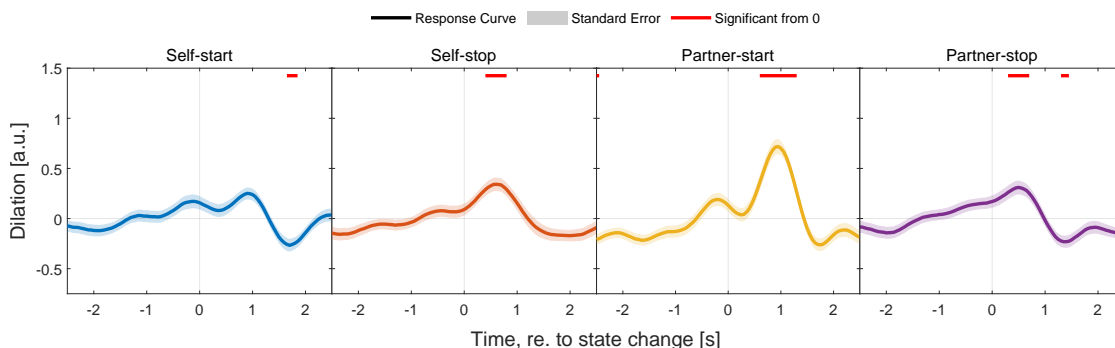


Figure 2.3: State change response curves obtained by averaging the results obtained for all conversations. The highlighted regions indicate the pointwise standard errors of the mean. The red bar indicates intervals where the resulting p-value from a pointwise one-sample t-test comparison with 0 is less than 0.1.

2.3.2 By condition

To assess how the pupil responses may vary based on expected difficulty of the conversation, the state change responses can be found by averaging only across conversations that took place in the same conditions. For this analysis, we compared the results between the quiet and 70 dBA noise condition, with the expectation that this combination will have the highest disparity in perceived difficulty and therefore emphasize any processing differences that exist.

Figure 2.4 shows the conditional state change responses along with the segments within which the two curves are significantly different from each other ($p < .01$). We see significant systematic differences by condition around all four state changes. We see significant differences in pupil response after the state change has occurred in all four cases. Significant segments are observed before the state changes corresponding to a talker’s own turn, and after the state changes in all four curves.

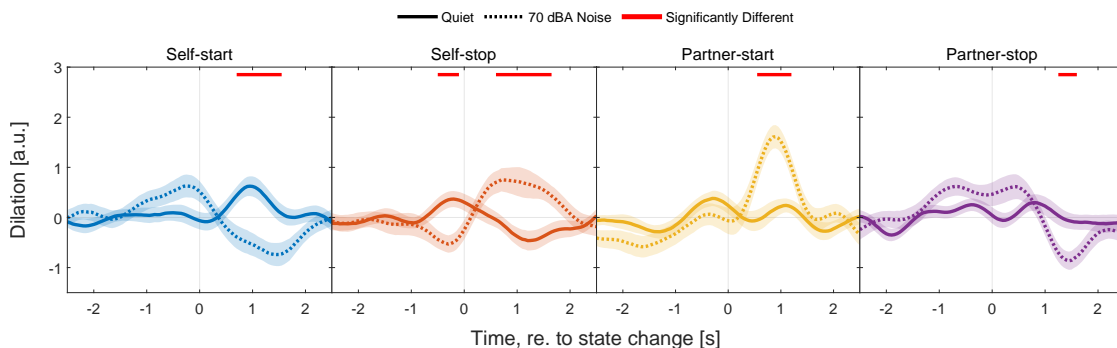


Figure 2.4: State change responses obtained only by averaging conversations that took place in quiet (solid line) and 70 dBA noise (dotted line). The highlighted regions indicate the pointwise standard errors of the mean. The red bar indicates intervals where the resulting p-value from the two-sample t-test comparing the two curves is less than 0.1.

2.3.3 By turn duration

To emphasize processing related differences in the state change responses based on turn duration, the extremes of the distribution are chosen as the bounds for short and long turns. Short turns are defined as being shorter than 500 ms, to capture brief one- or two-syllable utterances. Long turns are classified as being longer than 2300 ms, a boundary that was chosen to approximately match the same number of turns between the short and

long groups, with each containing $\sim 29.8\%$ of the total turns.. Note that the self-start and self-stop were classified based on the duration of a talker’s own turns, and partner-start and partner-stop were classified based on the duration of the partner’s turns.

Figure 2.5 shows estimates of distributions of turn duration by condition, with the regions corresponding to short and long turns indicated. As can be seen in Figure 2.6, significant differences in pupil response to all four state changes are observed based on the length of the turn.

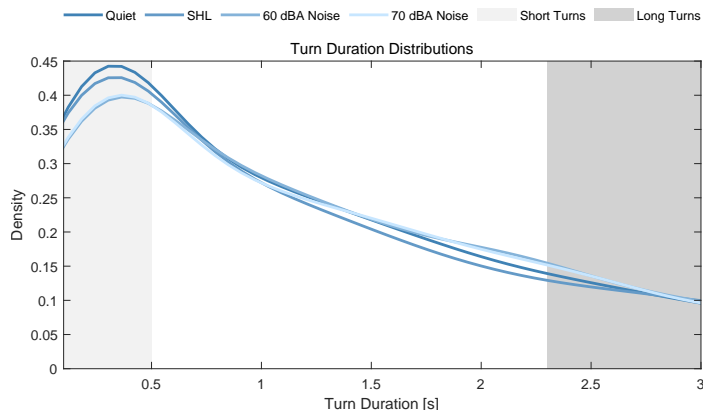


Figure 2.5: Distributions of turn duration for different conditions. Color indicates the condition the conversation took place in. The light and dark gray shaded regions represent short ($< 500\text{ ms}$) and long ($> 2300\text{ ms}$) turns, respectively. Each of the shaded regions contains approximately 29.8% of the total turns.

Due to the definition of long turns being nearly at the analysis window boundary, the state change responses are also derived using a window size of $\pm 5\text{ s}$ for the state changes corresponding to plots of short and long turns. The results can be seen in Figure 2.7, where it is observed that there is only one additional significant interval, in the self-start response curve, beyond the original $\pm 2.5\text{ s}$ window size.

2.3.4 Comparison to grand averaging

As mentioned previously, a common analysis method for task-evoked pupil response is to average the pupil’s dilation trace over many individual trials. This approach can be applied to our case by averaging across fixed windows around every state change, which we will call a grand average, with cases where the window would extend beyond the bounds of conversation excluded.

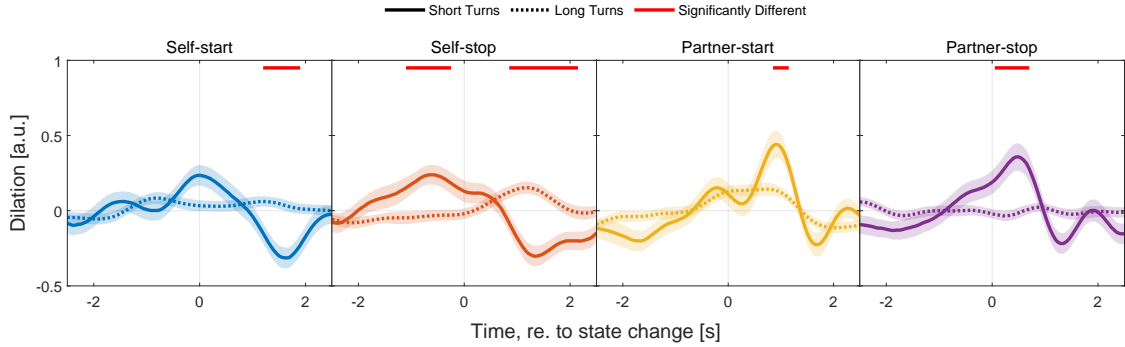


Figure 2.6: State change responses corresponding to turns of different durations. The solid line corresponds to short turns (<500 ms) and the dashed line to long turns (>2300 ms). The highlighted regions indicate the pointwise standard errors of the mean. The red bar indicates intervals where the resulting p-value from a pointwise two-sample t-test comparing the two curves is less than 0.1.

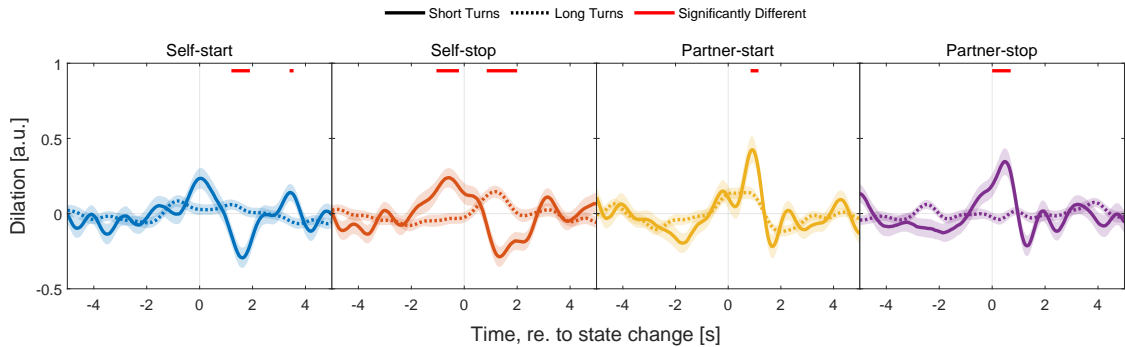


Figure 2.7: State change responses corresponding to turns of different durations derived using a ± 5 s window size. The solid line corresponds to short turns (<500 ms) and the dashed line to long turns (>2300 ms). The highlighted regions indicate the pointwise standard errors of the mean. The red bar indicates intervals where the resulting p-value from a pointwise two-sample t-test comparing the two curves is less than 0.1.

Figure 2.8 plots the previously derived response curves, from Figure 2.3, against the curves obtained from a grand average of pupil responses around every state change, across all participants and conversations. Due to the differences in scaling that happens because of regularization, both sets of curves are normalized such that the set of all TRFs has zero mean and unit variance, and the set of all grand average curves has zero mean and unit variance.

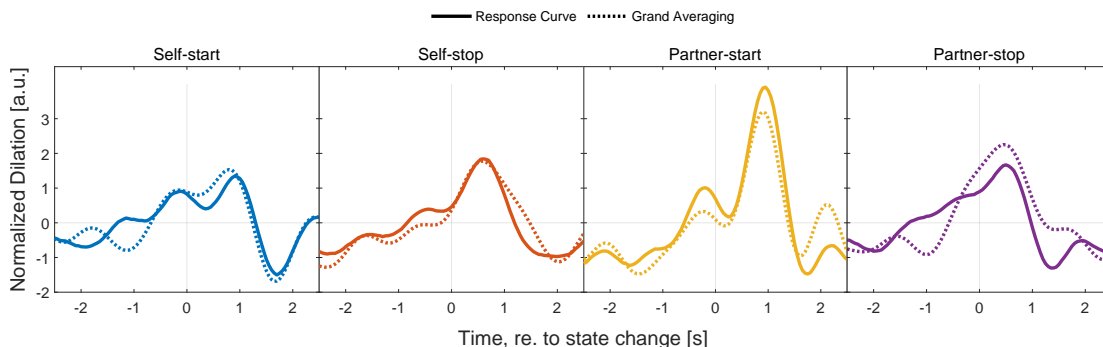


Figure 2.8: Derived state change responses plotted against the grand averages of fixed windows. The solid lines correspond to the derived pupillary state change responses, as in Figure 2.3. The dotted line is the result of averaging a fixed window size around every occurrence of each state change.

2.3.5 Analysis of gaze correction

In the following analysis of gaze correction, we assume that a gaze depth below the previously defined threshold indicates a talker looking at their image, and a depth beyond the threshold indicates a talker looking at their partner. Given that the Diapix task requires participants to predominantly look at the image to complete the task, we think this is a reasonable assumption.

Estimates of the distributions of gaze depth and duration of fixations to the partner, for each condition (i.e. pooled across replicates and participants) are plotted in Figure 2.9, revealing that most of the time participants looked at the image, and that glances at their partner are typically short (the mode of the distribution was 70 ms). Also displayed are the total percentages of each conversation that were spent looking at the partner. It is observed that in all four conditions, the median percentage is less than 5%, confirming that, on average, people spend most of each conversation looking at the image.

Analysis of the pupil responses to a changing gaze target reveals significant effects on pupil dilation, as seen in Figure 2.10. There are also short segments where the pupil response curves to change of gaze are different between quiet and noise, as seen in Figure 2.11.

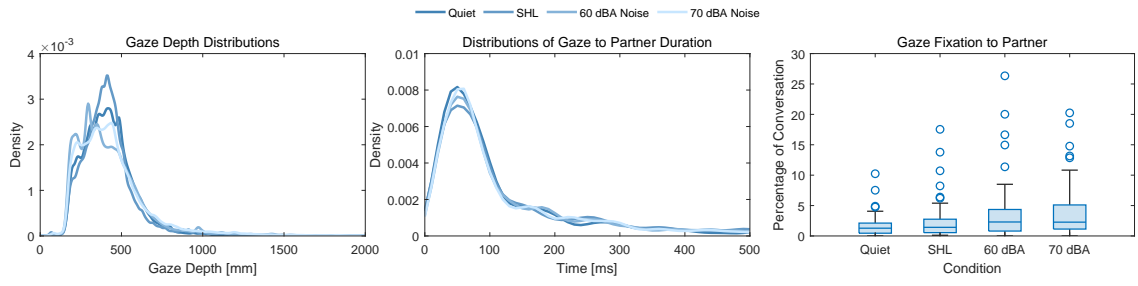


Figure 2.9: Distributions of gaze depth and the duration of fixations to the partner across all conversations, by condition. Color indicates the condition the conversation took place in. Also included are the overall percentages of each conversation that were spent looking at the partner, by condition, determined as the percentage of time points where the gaze depth was greater than the 95 cm threshold defined previously

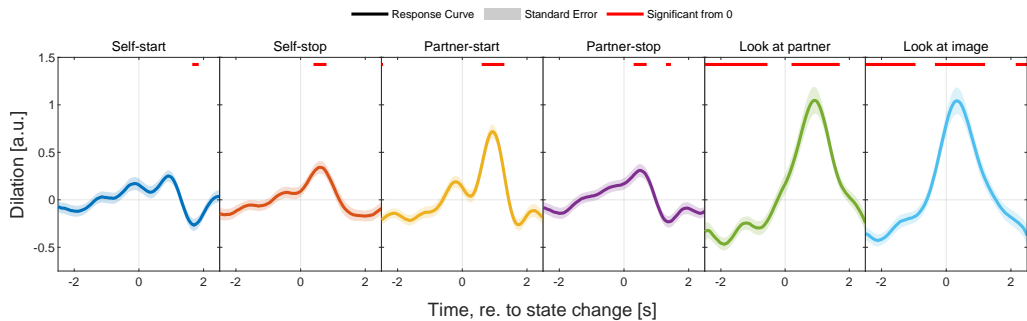


Figure 2.10: State change responses found by averaging across all conditions, including plots of pupil response to change of gaze target. The state-change responses are identical to those in Figure 2.3. The highlighted regions indicate the pointwise standard errors of the mean. The red bar indicates intervals where the resulting p-value from a pointwise one-sample t-test is less than 0.1.

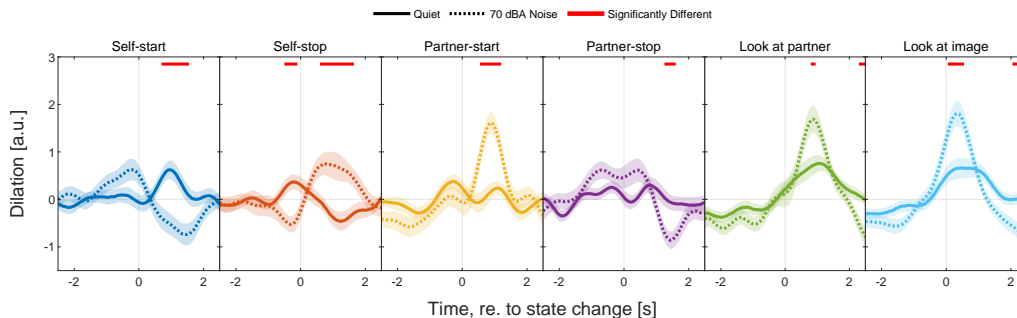


Figure 2.11: State change responses by condition, including plots of pupil response to change of gaze target. The state-change responses are identical to those in Figure 2.4. The highlighted regions indicate the pointwise standard errors of the mean. The red bar indicates intervals where the resulting p-value from a pointwise one-sample t-test is less than 0.1.

2.4 Discussion

2.4.1 Interpretation of state change response curves

In this work, we present a method for estimating how pupil response varies around turn-taking in conversation. This method not only estimates the pupil response to specific conversational state changes (e.g., a partner starting their turn), but also considers the other state changes and gaze behavior while doing so. Importantly, we found significant systematic pupil responses to turn-taking in conversation that varied based on background condition and the duration of the turn.

The significant portions in the response curves found by averaging across all conversations seem to correspond with our expectations around effort in conversation. Most notably, there is a large pupil dilation that peaks ~ 1 s after a partner begins their turn, which may be indicative of directing effort towards listening. It is possible that this could also be indicative of arousal as a reaction to hearing a new auditory stimulus. If this is the case then there should be no difference based on the background condition of the conversation, or the duration of the turn. However, when deriving the state changes only for conversations in alike conditions, this increase in dilation was found to be much larger in noise than in quiet, but rather of task demands via increased noise level, which suggests increased effort investment. Similarly, a larger dilation was found to occur as a reaction to a short turn than a long one. It is unclear exactly why a short turn would exhibit a larger pupil response than a long one. One possible explanation is that dilation responses

to short turns are quicker and more consistent, and therefore the peak of the dilation is less temporally smeared during the regression and subsequent averaging across conversations.

Another observation from the curves derived across all conversations is the significant dilation after a person stops talking. This effect is also likely related to the reallocation of effort towards listening. Intuitively, this effect should be attributed to both the end of a talker’s turn and the start of their partner’s turn, as it is unlikely that devoting effort towards listening is purely reactionary to a partner starting speaking. It is instead expected that talkers would begin reallocating effort toward listening as they finish their speech production and planning. However, the interval to the start of the subsequent partner’s turn, the FTO, can exhibit large variability, and therefore the timing of the preparatory effects may be more highly correlated with the end of the preceding turn rather than the start of the following turn. A similar dilation is observed shortly after the end of a partner’s turn, which could be related speech planning. Notably, though, there seems to be a lack of indication of effort prior to beginning one’s own turn. Previous studies of neural correlates of turn-taking have shown results that suggest speech planning starts well before a conversational partner’s turn ends, and begins as soon as it can based on the content of the partner’s speech (Bögels, 2020; Bögels et al., 2018). This observation could support the hypothesis that an increase in effort directed toward speech planning may be more temporally correlated with the end of a partner’s turn, as the point when planning can begin will depend on their speech, and not your own. Another possible explanation for the lack of response to speech planning observed in the self-start curve could be due to the task. Perhaps in the Diapix task, less sophisticated speech planning is required than in more typical conversations, as responses are usually either describing what you see, or replying to a partner’s description of what they see. The latter of which could often be spontaneous. For example if a partner is describing a portion of the scene, and suddenly describes something you don’t see, that would prompt a quick response that you may not have necessarily been planning for.

Further differences were found between the response curves derived based on the condition of the conversation. The pupil generally seems to constrict as one starts speaking in noise but dilates as one starts speaking in quiet. We suggest that this effect is related to an increase in effort required to listen and comprehend in noise resulting in a relative decrease in pupil size when going from listening to speaking. This suggestion is supported by the findings from Li et al. (2020), which showed that pupil size while listening during conversations with a higher communication load was larger than in conversations with a lower communication load. Although communication load in this context was not increased via background noise, both have the effect of introducing difficulty into the conversation. To further support this suggestion, a post-hoc statistical analysis was performed on the

participants’ pupil diameters during different phases of the conversation. The mean pupil diameter during speaking and listening time was computed for each talker in each conversation. A pair of linear mixed effects models of the form Mean Diameter \sim Action + (1 | Talker) + (1 | Replicate) was fit to the mean diameter measurements from the quiet and 70 dBA noise conditions, separately, where ‘Action’ was a categorical variable that specified whether the mean diameter value corresponded to speaking or listening time. Analyses of variance revealed that pupil size was significantly larger while listening than speaking in noise [$F(1, 74) = 25.11, p < .001$], but that there was no significant difference in pupil size between speaking and listening in quiet [$F(1, 70) = 0.26, p = 0.61$]. These results support the previous suggestion that the changes in pupil size observed in the TRFs could be related to differences in effort between speaking and listening in the more difficult condition. A similar but opposite effect is observed when one stops talking. The pupil dilates in noise while preparing/starting to listen and does not in quiet, likely for the same reasons.

The duration of a turn probably plays a key role in the amount of effort required, both in speaking and listening. For example, the amount of speech planning required for a simple one-syllable affirmative response is likely less than a more complex sentence, such as describing some region of a picture. The results presented here show that pupil responses to short turns are generally larger and more dynamic than pupil responses to longer turns. However, this observation could also be an artifact of the window size used, given that the long turns were defined as being at least 2.3 s, all long turns should last until nearly the end of the window, or beyond it. Analysis using a longer window size of 5 s in either direction revealed one further interval, beyond 2.5s, where the responses to long and short turns were significantly different, which was in the self-start response curve as seen in Figure 2.7. This suggests that it is not a result of the window size, but likely some other effect. One possible suggestion is that during a long turn, there will be effort required at variable points throughout the turn. Therefore, when averaged across many long turns and across different people, these effects are not as temporally consistent as the response to a short turn. This could explain why, for example, the self-start and partner-stop responses to long turns are nearly flat, whereas those responses to short turns are not. One possible justification for this can come from the previously mentioned tendency of talkers to start planning their turn as soon as they are able (Bögels, 2020). In longer turns, the point at which planning can begin will be more variable, as it can occur over a larger range of time. Therefore, the effects related to speech planning will be less temporally consistent. From analyzing pupil response to short turns, we can see how the pupil responds as attention changes happen more quickly. For example, it is clearly visible that a person has stopped talking by the end of the self-start curve corresponding to short turns.

However, it should also be noted that there is likely to be significantly different pro-

cessing demands based on the content of speech, and not just the duration. Although in some cases the duration of an utterance may indicate its content, e.g., if an IPU is less than half a second it is more likely to be a yes/no response than a descriptive sentence, classifications about content cannot be definitively made from duration. Content analysis was excluded from this work for two main reasons, the first is that the speech signals could not be shared due to the guaranteed anonymity in the informed consent letters signed by participants. The second is that content analysis tends to require a significant amount of manual labelling, which may not be desirable in future applications of this work, where clinicians or engineers would be looking to evaluate hearing loss or the performance of hearing aids in an automated fashion. However, given recent advancements in machine-learning language models, it's possible that automated content analysis could be looked at in future works to provide more insight into the differences in cognitive processing between different types of turns.

2.4.2 Advantages of the proposed method

One of the proposed advantages to using the regression-based method over simply averaging a window around every turn-taking event was the ability to separate (i.e., demix) the effects of each state change on the pupil response. The results in Figure 2.6 suggest that the modeling approach used here does have a demixing effect, as we can see the peak that our model has attributed to partner-start appears to be showing up, although temporally smeared, in the curves belonging to all the other state changes. This is visualized by an increase in amplitude and a broadening of the dilations in the other three grand-averaging curves. Another proposed advantage of the regression method is the ability to account for covariates in the model, which helps prevent misattribution of pupil response to, for example, looking at your partner to one of the state changes. This is a valid concern, as gaze plays a significant role in turn-taking behavior (Degutyte and Astell, 2021). Our findings revealed significant responses of the pupil to a change of gaze target. The response curves corresponding to the change of region of interest indicate a dilation after a talker looks toward their partner, and a constriction after they look back at the image, aligning with the expected effect of the near/far pupil response (Kasthurirangan and Glasser, 2005). This result suggests that the model is accounting for differences in gaze behavior, when deriving the effects of turn-taking. Interestingly, different responses are observed to a change of gaze target in quiet versus noise. One possible explanation for this difference is the purpose of glances at the partner. In noise, the change of gaze target may be effort related, whereas in quiet they may more often be used as a social cue. If people are looking at their partner in noise when they are experiencing difficulty in the conversation, it

would make sense that the dilation would be stronger, as it would be temporally correlated with both a distance-based pupil accommodation and an increase in effort. Whereas in quiet, there may be no effort related effect that covaries with the near/far pupil response. This idea can be partially supported by previous studies of gaze behavior in conversation, which have shown that talkers tend to spend more time looking at their conversational partner’s mouth when conversing in noise than in quiet (Hadley et al., 2019). This implies that when conversational difficulty increases, talkers are relying upon the long-studied benefits of visual information for increased speech intelligibility Erber (1975); Sumbly and Pollack (1954). However, the previously mentioned role of eye-gaze in regulating turn-taking behavior (Degutyte and Astell, 2021) would suggest that talkers also direct their gaze toward their partner in quiet, where it is less likely that the talkers would be doing so for the benefits of visual information on listening.

Although there are interesting results and potential interpretations related to change of gaze target, most of the time, people looked at the image rather than their partner in all conversations. Across all conditions and conversations, the median percentage of time each participant spent looking at their partner was less than 5%. This is likely an effect of the task used to elicit the conversations, as Diapix necessitates near constant reference of the image to be able to determine if differences exist. For this reason, it is likely that accounting for the gaze behavior in this experiment had little effect. Although in more naturalistic communication settings, such as a free conversation, the role of gaze, and the correction thereof, would likely be more important.

2.5 Conclusions

This chapter presented a method for deriving pupillary responses in interactive conversations based on turn-taking. This approach revealed systematic pupil responses to turn-taking in conversation, which can potentially be used to infer how cognitive effort varies during the transition from listening to speaking in conversation, and how these effects change based on background condition or turn duration. In addition to being applicable to interactive conversation, advantages of this method over an epoch-based averaging approach are the ability to include external covariates in the model (such as gaze) rather than having to process pupil data to account for these effects, and the capability of demixing overlapping responses from neighboring events. Further work is needed to understand how the interpretation of the results here can provide insight into changes in processing demands during conversation. A potential follow-up would be to have a listener-observer follow along while talkers participate in a subset of the Diapix tasks and measure their

pupil response. This would separate out the cognitive effects from speaking and those associated with task demands and listening. Results from such a study would provide further insight into how attention is divided, and effort is reallocated during conversation.

Acknowledgments

This work was funded by the William Demant Foundation (21-2520), and completed collaboratively with Eriksholm Research Centre.

Chapter 3

Methods for manipulating conversational dynamics in near real-time

3.1 Introduction

Near real-time (also sometimes called ‘online’) audio processing typically has a high barrier to entry, requiring either extensive knowledge of computationally efficient programming languages, such as C++, or access to purpose-built low latency data acquisition systems, such as those built by National Instruments. For many psychological and behavioral researchers, these systems may be inaccessible due to having different technical backgrounds. Therefore, it is likely that auditory research as a whole would benefit from an accessible yet performant customizable audio processing system.

Additionally, current conversational dynamics research is limited to testing hypotheses that can be evaluated through environmental changes, such as the introduction of background noise. However, the ability to modify the dynamics of communication in real-time would enable the design of an entirely new set of experiments. Further, the development of such a system is necessary for the experimental design proposed in Chapter 4, where the goal is to manipulate the timing of talkers’ turn-taking in conversation and observe any resulting adaptive behavior.

This work presents a low cost, latency, and barrier to entry method for online audio processing that can be implemented on any computer with any digital audio interface.

3.2 Building a processing system

This section outlines the design and development of a Python based audio processing system, starting with the requirements and constraints of said system. The tools that make up the system are then discussed, followed by their integration into a cohesive program. Finally, a set of example processing modules is introduced to demonstrate the system’s capabilities.

3.2.1 Design requirements and constraints

First, the system needs to be able to read and write audio via a digital audio interface. The developed system should also be able to implement audio processing on blocks of audio as they pass from input to output. Depending on experimental design, this processing should either be able to on every incoming block of audio, or at varying intervals of time. Parameters of the processing taking place should be able to be either fixed or adaptive. Although one goal is to perform some online processing, this will typically need to be accompanied by post-experimental data analysis. Therefore, the system should also be able to perform data acquisition (i.e., saving input signals to an audio file) without interfering with the audio processing taking place.

Another key consideration is the latency of the system. Latencies as low as 6 ms have been shown to be “disturbing” for hearing aid users receiving auditory feedback of their own voice through open-canal fittings (Stone et al., 2008). Occluded fittings, however, resulted in a higher tolerable delay of 15-30 ms (Stone and Moore, 2005). For conversation, while the constraint of low latency for own voice feedback remains, much higher latencies for receiving the signal from a partner can be tolerated. Many modern digital communication systems, such as video conferencing services, have latencies in the hundreds of milliseconds (e.g., 300-1000 ms over Zoom (Boland et al., 2022)). Nevertheless, it is desirable to minimize system latency as it is likely experiments involving this system would include control conditions where differences from a natural communication environment are minimized.

Many factors impact the latency of a digital audio system. The audio interface used will inherently have some input and output latencies, and audio is typically acquired in blocks, which are a set of a fixed number of contiguous samples, which also introduces some amount of latency. As an example, for a typical audio system operating at a 48 kHz sample rate, a buffer size of 128 samples introduces just over 2.5 ms of latency into the system, plus the inherent input and output latencies of the hardware being used.

In summary, the goal of this work is to design a framework that simultaneously streams audio to and from a digital audio interface, performs some processing on the streamed audio, and saves the audio recordings. The system should also be able to perform all these operations with minimal latency.

3.2.2 Tools

The Python SoundDevice package is a set of Python bindings for the popular open-source C/C++ PortAudio library and can be used to read and write audio block-by-block from a digital audio interface. SoundDevice also supports the ASIO and CoreAudio protocols, enabling the development of cross-platform applications that utilize low latency and high fidelity audio input-output (IO) streams. SoundDevice has the ability to open input, output, or simultaneous IO streams, which then provide blocks of audio to Python in the form of NumPy arrays. The size of the audio blocks is determined by the buffer size of the audio interface being used, which is typically a power of two in the range of 32 to 1024 samples. SoundDevice also executes a callback function on each block of audio it acquires. This callback can include any desired block-by-block processing, such as simply passing the audio through to the output, applying a filter, monitoring and manipulating communication behavior, or a seemingly infinite number of other processing algorithms. Additionally, having a separate data acquisition system running, e.g., a sound recording application such as Audacity, could affect performance because both applications would access the incoming blocks of audio from the interface. For this reason, SoundDevice also provides the ability to write the incoming audio to any of the standard audio file formats.

Typically, a Python program only operates on a single thread which is acceptable for the case where our audio processing might only contain a callback function, such as implementing a simple digital filter on each incoming block of audio. However, in a case where we want some other tasks running at a different rate than every block, we can utilize the Python *threading* library, enabling a pseudo-multithreaded application, which will execute our threaded function at some parametric fixed interval of time. This could be used, for example, to update the coefficients of a filter based on some characteristics of a time history of the input signal.

It may also be desirable to have some parameters of the processing algorithm be adjustable while the program is running. Although this could be done through a command line interface, it could be beneficial to have a graphical display to keep track of the current parameters, and enable easier adjustment of multiple parameters. Additionally, one may want to plot some feature of their signal, such as a spectrum, or display an updating value,

such as a sound level. To enable this, a graphical interface can be designed in QtDesigner and converted to a Python program using the PyQt6 library.

These tools can interface with each other, as illustrated in Figure 3.1, in one cohesive Python program.

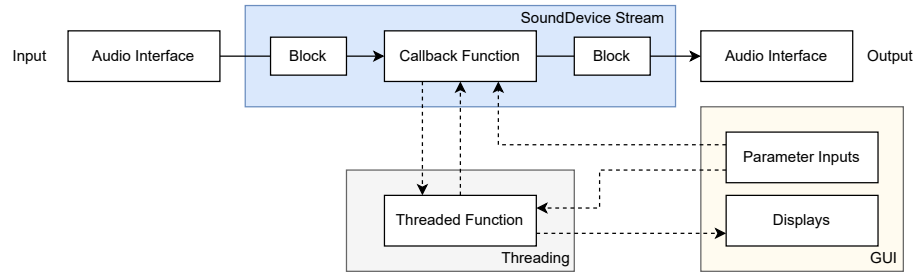


Figure 3.1: A block diagram for an audio processing system that can read and write audio from an audio interface using SoundDevice, execute functions at a different rate than the stream rate using threading, and pull parameters or display outputs using a graphical user interface built with Qt.

3.2.3 User interface

The graphical user interface (GUI or UI) was designed to include dropdown boxes to select the input and output device, IO channels, number of channels (feedback or dyadic), and a processing module. Figure 3.2 displays the designed UI for (a) general auditory feedback and (b) dyadic experiments, and also (c) an example UI that includes processing options for a study involving delaying responses in conversation, which is discussed in more detail later.

The objects in the UI can be accessed through the PyQt6 package, such that when the SoundDevice stream is started, it will use the options selected in the UI. Channel selection can also be changed when a stream is already running, along with any parameters necessary for the specific processing module.

3.3 Designing processing modules

To enable different types of experiments, a Python class can be written that contains the callback and threaded functions, which we will refer to as a module. The class is

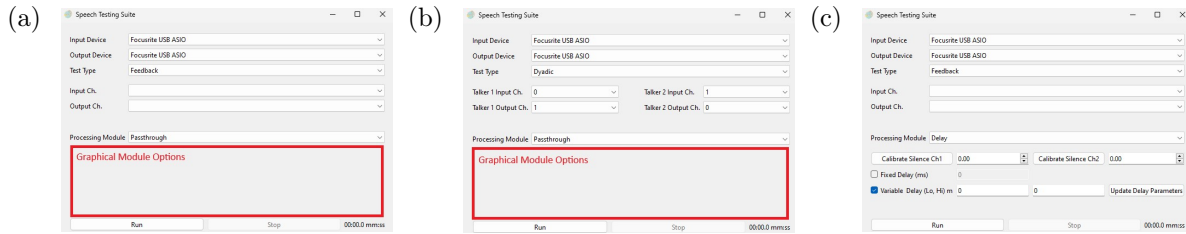


Figure 3.2: Examples of a user interface developed for (a) general auditory feedback experiments, (b) general dyadic conversational experiments, and (c) a dyadic conversational experiment involving delaying responses from one talker. The red boxes in (a) and (b) indicate the area where processing specific parameters or displays could be included, as demonstrated in (c).

instantiated upon selection from the module processing dropdown box. There are only 3 required functions for a processing module to operate within this framework: *start()*, *stop()*, and *callback(data)*. The *start()* and *stop()* functions should require no arguments. The *callback(data)* function should accept one argument, *data*, which is a [b,n] array, where *b* is the block size being used and *n* is the number of channels (1 for feedback, 2 for dyadic). Note that these functions are in addition to the required `__init__()` method when defining a Python class.

3.3.1 A callback only example

In the simple case that only a callback function is desired, we can design a class as follows. In this example, we want to attenuate our incoming signal before outputting it. We can generalize the module to apply to both feedback and dyadic experiments by ensuring that the attenuation happens in an element-wise manner. A block diagram for the system to be described is displayed in Figure 3.3. The corresponding Python class is shown in Figure 3.4, for reference.

We start by initializing our processing class, named *StaticGain*. During initialization, we pass to the module the higher level window and stream arguments, which correspond to the UI and `SoundDevice` stream objects, respectively. Although this is not necessary in this example, it can be useful if, for example, the processing being performed is dependent on some parameter of those objects, such as the sample rate. We can then initialize a gain parameter, which will provide the scaling for our signal. In this case, the gain is defined in dB, as -3 dB. However, the easiest implementation of attenuation will be to simply scale our input signal, therefore we convert the gain to linear units.

Following this, the `start()` and `stop()` functions are defined. In this case, the start and stop functions are simply used to print to the console an indication that the module is running. In some cases, for example if the processing parameters adapt over time, these functions could be used to reset the parameters, or save some information about the processing that took place.

Finally, the callback function is defined. In this simple example, all that is required is to scale the input signal, which is passed to the callback function as `data`. We can simply multiply the signal by the linear gain value, as the product is performed element-wise by default.

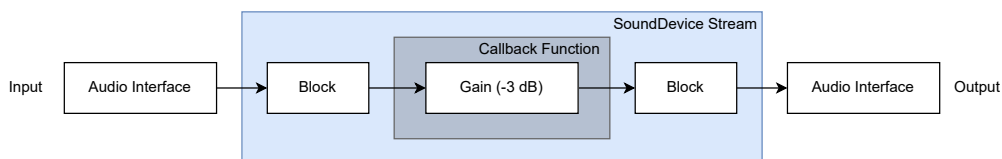


Figure 3.3: A block diagram for a processing module that applies a static amount of gain to an input signal before outputting it.

3.3.2 A threaded example

When designing a processing module that will include threading, there are a few other important considerations. First, the module class should inherit from the `threading.Thread` parent class. There are also some additional statements needed in the initialization function, such as the definition of the stop event, which stops the threaded process, and sleep period, which determines how frequently the threaded function is executed.

Additionally, the previously mentioned necessary `start()` function should be renamed to `run()` due to the existence of a `start()` method within the `threading.Thread` parent class. The `run()` function is executed when the existing `start()` function is called at the start of an audio stream, without explicitly calling it. Therefore, threaded and callback only modules can be used interchangeably within the framework.

In this example, we adapt the previously defined gain function to adjust the amount of gain every 30 seconds by -3 dB. Therefore, the current gain will be a function of time. For reference, a block diagram for the system to be described is shown in Figure 3.5, and the corresponding Python code in 3.6.

```

class StaticGain:
    """An example module demonstrating a simple callback function that applies gain to the input signal
    ↪ before outputting it"""
    def __init__(self, win, stream):
        self.win = win
        self.stream = stream
        # Can also define parameters during initialization
        self.gain = -3 # Apply a -3dB gain to the signal
        self.gain_linear = np.power(10, self.gain/10) # Convert gain to linear units

    def start(self):
        """Function that's executed upon starting an audio stream"""
        # This function may need to include resetting some parameters, if they are adaptive over the course
        ↪ of a trial
        print("Starting StaticGain Module")

    def stop(self):
        """Function that's executed upon stopping an audio stream"""
        # This function may need to include saving additional information from the experiment
        print("Stopping StaticGain Module")

    def callback(self, data) -> np.ndarray:
        """Audio processing callback that's executed on each block of incoming audio"""
        # Can insert any block-by-block processing here. In this case we just apply gain to our input
        return data * self.gain_linear

```

Figure 3.4: Example Python code for a processing module that applies a static amount of gain to an input signal before outputting it.

Initialization of the class is similar to the static gain module, but we additionally define the stop event and sleep period. We have also written a function to convert the gain dB to linear units, as this operation will need to happen repeatedly.

In the *run()* function (equivalent to *start()* from above), a while loop is included to execute the threaded function after the sleep period has passed. The threaded function in this case is *adapt_gain()* which subtracts 3 dB from the current gain value, and then converts the current gain value to linear units. The *stop()* function, ends the threaded process. The callback is identical to the previous static gain module.

3.3.3 Integrating the modules into the framework

After the modules themselves are developed, they can be relatively simply integrated into the main framework. In it's current state, the code of the main framework needs to be modified to include more modules (which is contained in the *testing_swite.py* file). However, the eventual goal is to be able to include these classes as separate files in a

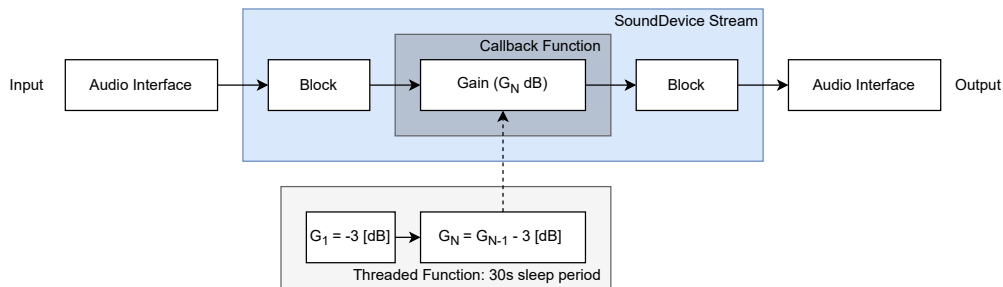


Figure 3.5: A block diagram for a processing module that applies a static amount of gain to an input signal before outputting it.

modules subdirectory, and have the framework import all processing modules accordingly.

First, an item needs to be added to the dropdown box for each of the developed modules, which is done during initialization of the program. Second, a statement that will instantiate the processing module upon selection needs to be added. These modifications can be added as shown in Figure 3.7. Provided that the module contains the necessary *start()* (or *run()*, if threaded), *stop()*, and *callback(data)* functions, when 'run' is clicked in the UI, an audio stream will begin that is performing the processing included in the selected module.

If a UI element is desired for the developed module, one can be made in QtDesigner, and added as an extra page within the *module_options* stacked widget layout. In this case, the order of modules placed in the dropdown is important, as the page displayed in the UI will correspond to the selected index of the dropdown box.

3.3.4 A more complex example

The original intention behind developing this framework was to enable an experiment that modified the floor transfer offsets of a conversation in real time. In order to do this without audible effects from adding or removing delay, the state of the conversation must be tracked, and changes to the amount of delay on a talker's communication line must only be made when that talker is not speaking. Therefore, the module developed to carry out this experiment had multiple simultaneous processing components, as delay needed to be manipulated and the state of the conversation needed to be tracked. However, state tracking should be conducted on a near real time basis and not at a fixed interval of time. Therefore, both processes can be implemented in a single block-by-block callback, rather than using threading. In order to accurately assess the state of the conversation, a time

```

class AdaptiveGain(threading.Thread):
    """An example module demonstrating the use of threading by decreasing the amount of gain applied to a
    ↪ signal every 30 seconds"""
    def __init__(self, win, stream):
        super(AdaptiveGain, self).__init__()
        self._stopevent = threading.Event()
        self._sleepperiod = 30 # seconds
        self.win = win
        self.stream = stream
        # Can also define parameters during initialization
        self.gain = -3 # Start with -3dB gain to the signal
        self.gain_linear = None # Initialize this with no value, it is defined from the next function call
        self.gain_to_linear()

    def gain_to_linear(self):
        """Converts gain from dB to linear units"""
        self.gain_linear = np.power(10, self.gain / 10) # Convert gain to linear units

    def adapt_gain(self):
        """Decreases the current gain value by 3 dB, this is the threaded function"""
        self.gain -= 3 # Decrease by 3 dB every sleep period
        self.gain_to_linear()

    def run(self):
        """Function that's executed upon starting an audio stream"""
        # In this case we use run() instead of start() due to the threading parent class
        print('Starting AdaptiveGain Module')
        try:
            self._stopevent.clear()
            while not self._stopevent.is_set():
                self.adapt_gain()
                if self._stopevent.wait(timeout=self._sleepperiod):
                    break
        except Exception as e:
            print(e)

    def stop(self):
        """Function that's executed upon stopping an audio stream"""
        # This function may need to include saving additional information from the experiment
        print("Stopping AdaptiveGain Module")
        try:
            self._stopevent.set()
        except Exception as e:
            print(e)

    def callback(self, data) -> np.ndarray:
        """Audio processing callback that's executed on each block of incoming audio"""
        # Can insert any block-by-block processing here. In this case we just apply gain to our input
        return data * self.gain_linear

```

Figure 3.6: Example Python code for a processing module that applies an adaptive amount of gain to an input signal before outputting it, using threading to adapt the amount of gain ever 30 seconds.

```

class Window(QMainWindow, Ui_MainWindow):
    def __init__(self, parent=None):
        ...
        # Add options for the new modules to the processing module dropdown box
        self.combo_proc_module.addItem('Static Gain')
        self.combo_proc_module.addItem('Adaptive Gain')
        ...

        ...
    def update_module_options(self):
        """Loads the selected processing module"""
        ...
        # Upon selection of one of the new modules, load that class as self.Module
        elif self.combo_proc_module.currentText() == 'Static Gain':
            self.Module = static_gain.StaticGain(self, self.stream)
        elif self.combo_proc_module.currentText() == 'Adaptive Gain':
            self.Module = adaptive_gain.AdaptiveGain(self, self.stream)
        ...

```

Figure 3.7: Additions to the main framework program that must be made to include a new processing module.

history of the audio signals is necessary, as the contiguous duration of speaking and silence by each talker needs to be assessed to determine when floor transfers occur.

For this reason, a pair of processing classes were developed. The first managed the amount of delay on the communication lines. The second tracked the state of the conversation. Given that the ability to manipulate delay without audible artifacts was dependent on the state of the conversation, the state tracking class can be instantiated within the delay module class. Therefore, from the framework’s perspective, this still only operates as a single processing module. With this design, there is also the advantage of being able to implement the same state tracking class within other conversational processing modules, such that processing parameters could be adapted on a turn-by-turn basis rather than at some fixed interval of time, as would be the case with a threaded function.

A simplified block diagram of the delay processing module can be seen in Figure 3.8. More detail on the algorithm used to detect state changes and manipulate delay values is included in Chapter 4, and the classes used to manipulate delay and perform state tracking are include in Appendix A.

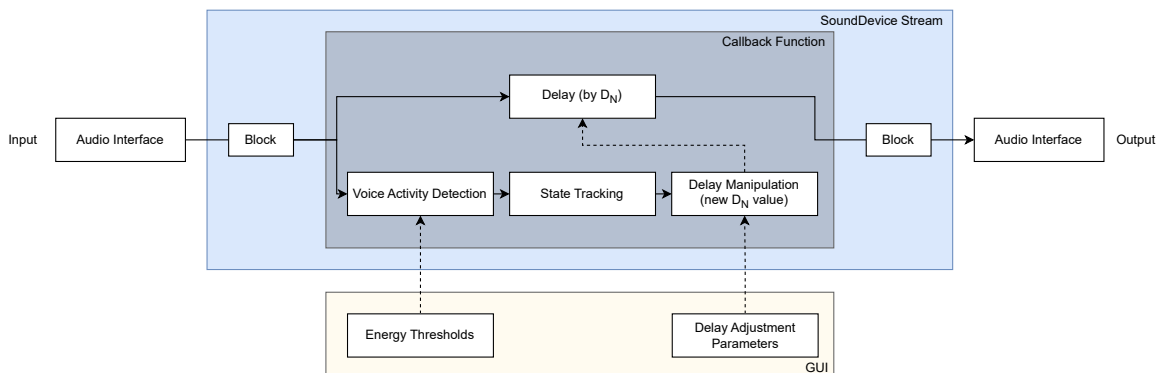


Figure 3.8: A block diagram for a processing module that tracks the state of a conversation and manipulates the amount of delay on the transmission line between talkers, when appropriate.

3.4 Performance

The performance of the system was evaluated with the delay module using audio interfaces of varying grades (entry-level consumer to high-level professional) while running on a laptop computer (Dell Latitude 7490 equipped with an Intel i7-8650u). The system experienced audio dropout on all interfaces at buffer sizes less than 128 samples, therefore testing was performed at this buffer size.

3.4.1 Round-trip latency

The round-trip latency (RTL) of the system was tested by inputting an impulse to one of the systems input channels (via a microphone in the form of a snap). This signal was then passed through the system with the output looped back into another input channel. The cross correlation was computed between the two input signals, and the lag at which the maximum correlation was achieved was defined as the estimated RTL. The results are displayed in Table 3.1, and demonstrate that latency decreases monotonically (although inversely) with cost, suggesting that lower latencies are achievable with higher-grade interfaces.

In all cases, the latencies are below the acceptable amount found by Stone and Moore (2005) for occluded hearing aids. Therefore, the system should be acceptable for altered auditory feedback experiments with a 128 sample buffer size, provided that closed-back

Interface	Round-trip latency	
	samples	ms
Focusrite Scarlett 2i2 Gen3	701	14.6
Universal Audio Volt 276	521	10.9
RME Fireface UFX III	325	6.8
RME Fireface UCX II	319	6.6

Table 3.1: Round-trip latency results for an impulse passing through the designed testing system for a variety of audio interfaces. All testing was done through the delay module, with no delay added, at a 48 kHz sample rate with 128 sample buffer size.

headphones are used. However, if timing is especially important in the experiment, it is likely that a lower latency interface is desirable.

One final note on latency is that there exists a trade-off between computational ability and latency. The delay program is relatively light-weight in terms of computation. If one needed to, for example, estimate formant frequencies, which requires more complex processing, then a larger buffer size may need to be used to enable more computational power. However, the idea with using threaded functions is that you can offload some of this processing such that it happens less frequently, and therefore places less strain on the system.

3.5 Conclusions

This work introduced a near real-time audio processing framework designed for use in conversational and auditory feedback experiments. The framework was developed in Python, such that there is less barrier to entry than other audio processing methods. Example modules were demonstrated to show that it is straightforward to build a custom process module, for example to apply and vary gain throughout an experiment. A more complex module was also introduced, demonstrating that the processing modules are not constrained to simply processing the input signal, but rather are to perform more demanding operations such as tracking the state of a conversation and varying a transmission line delay between talkers. Finally, the round-trip latency of the system was evaluated, and determined to be within the bounds of acceptable latencies for auditory feedback in occluded environments, even on the most entry-level system. Overall, this framework seems like a promising and useful tool for the future of auditory research, and the first experiment conducted with it is introduced in the following chapter.

Chapter 4

Sensitivity of talkers to the timing of turn-taking

4.1 Introduction

Conversation is a complex interactive activity involving not only speech production and perception, but also the interaction and adaptation of talkers to each other and to their environment.

Talkers adapt to challenging acoustic environments during conversation in a variety of ways. The well-known Lombard effect describes a phenomenon in which talkers adjust their vocal effort (e.g., by speaking louder) in the presence of background noise, effectively increasing the signal-to-noise ratio received by any listeners.

However, more subtle adaptations also occur, such as the leaning in of talkers toward a conversational partner when conversing in noise. When entire conversations take place in noise, interlocutors lean in enough to provide a signal-to-noise ratio benefit of up to 3 dB when sitting and up to 9 dB when standing (Miles et al., 2023). However, this effect has been observed even in cases where no tangible acoustic benefit is realized. Hadley et al. (2019) found that when the background noise level varied every 15-25 seconds in a conversation, a positional adaptation was still observed in the form of a leaning in, although at a much smaller magnitude, such that there was no acoustic benefit (an estimated received level increase of 0.01 dB per 1 dB of added noise). This observation suggests that talkers may make some adaptations as social indicators rather than as accommodations with a realizable benefit.

Other behavioral adaptations are made as well, such as the direction of gaze towards a partner’s mouth when listening becomes more difficult. The benefits of visual information for understanding speech in noise have long been studied (Erber, 1975; Sumbly and Pollack, 1954) and talkers have been shown to take advantage of visual information when interacting in difficult acoustic environments, adapting their gaze behavior by looking at a conversational partner’s mouth more often when conversing in noise (Hadley et al., 2019).

Talkers have also been found to adapt their turn-taking behavior in conversation in response to increased difficulty. Levinson and Torreira (2015) introduced the metrics of floor transfer offsets (FTO), defined as the amount of time it takes for one talker to begin their turn after (or before) another talker has ended theirs, interpausal units (IPU), defined as connected speech by one talker, and pauses, which are gaps in a talker’s speech, to characterize the dynamics of conversation. To classify the state of a conversation based on these metrics, some other considerations must be made. For example, it is worth defining a minimum IPU length, to avoid identifying non-speech acoustic bursts as very short IPUs, and minimum pause lengths, to avoid misclassifying stop-consonants within IPUs as pauses. Heldner and Edlund (2010) suggest defining these bounds as 90 ms for the minimum IPU length, and 180 ms for the minimum pause length.

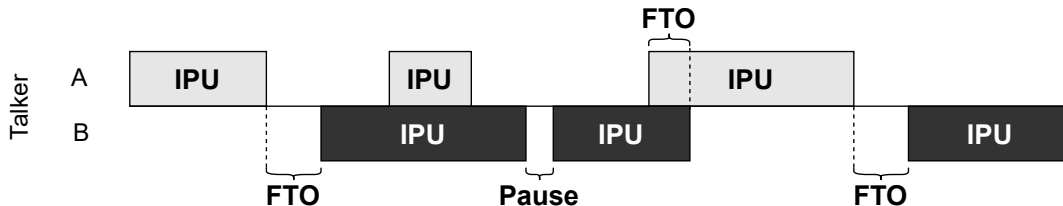


Figure 4.1: A sample turn-taking exchange between two talkers illustrating the definitions of and interactions between IPUs, FTOs, and pauses.

Previous studies have found that the durations of FTOs and IPUs increase when difficulty is introduced into a conversation (e.g., through background noise or a second language in Sørensen et al. (2021), or due to hearing impairment in (Sørensen et al., Submitted; Petersen et al., 2022)). These studies have also found that, in addition to an increase in the durations of FTOs, the variability of FTOs also increases. These observations have resulted in suggestions that increases in duration of FTOs and IPUs, and variability of FTOs can be interpreted as indicators of conversational difficulty level.

However, talkers don’t only adapt to their own difficulty, but also to the difficulty of their conversational partner(s). Previous work has shown that when one talker in a dyadic conversation received a distorted version of their partner’s speech, both talkers

exhibited altered speech production, suggesting that the talker receiving the unaltered signal is adapting to the increased difficulty of their partner (Hazan and Baker, 2011). Other studies have examined how speech production and communication behavior change when normal-hearing (NH) talkers interact with hearing-impaired (HI) talkers. NH talkers have been observed to adapt their speech, through increased level, mid-frequency emphasis, and formant frequencies in a manner that was correlated with the level of hearing loss of their HI partners (Beechey et al., 2020). Petersen et al. (2022) found that when NH and HI talkers hold conversations with and without hearing aid amplification, both talkers speak louder when the HI talker is unaided. However, this effect was only observed in conversations taking place in quiet and not in noise, perhaps suggesting that once the NH talker is experiencing difficulty (in the form of background noise), their sensitivity to the HI partner’s difficulty is diminished. Further to this, Sørensen et al. (Submitted) found that NH talkers exhibit significantly different adaptations of turn-taking behavior as an effect of noise when talking with HI talkers than with other NH talkers. For example, when comparing the FTO distributions of conversations taking place between two NH talkers (NH-NH) and an NH and HI talker (NH-HI), the NH-HI distribution is broadened and shifted to the right, indicating an increase in duration and variability of FTOs by the NH talkers in NH-HI relative to an NH-NH conversation. Thus, it seems that NH talkers are adapting their speech and behavior to an HI partner’s difficulty. However, it remains unclear how exactly the NH talkers are determining that their HI interlocutor is having trouble.

One hypothesis to explain this result is that the NH talkers are monitoring the timing of the HI talkers’ turn-taking, implying that the increase in magnitude and variability of the FTO is used as a cue by the NH talkers to infer that the HI partner is experiencing difficulty.

This study aims to examine if the timing of turn-taking by a conversational partner is used as a cue to infer that difficulty is being experienced. Specifically, we seek to determine if talkers recognize that their conversational partner is taking longer than expected to respond and modify their behavior in response. The expectation is that talkers will perceive an increased response time from their partner as an indicator of difficulty and modify their behavior in response, for example by increasing the duration of their IPU or decreasing the rate of speech, thereby providing the partner more time for speech comprehension and planning. To study this, we held interactive conversations between two NH talkers and simulated increases in the magnitude and variability of FTOs that has observed in more difficult conversations, without introducing any difficulty, by artificially delaying responses from one of the interlocutors. As a control to determine how these specific talkers adapt their behavior in a difficult environment, conversations also took place with and without

the presence of background noise.

To replicate the observed increase in magnitude and variability of the response time under difficult conditions, a system was built to implement and vary the delay on a communication line between talkers throughout a conversation. It is only necessary to delay one talker’s speech for the effect to be perceived by both talkers. This is demonstrated in Figure 4.2, where it is shown that the transmitted FTO (labeled as such because it is transmitted by the talker ceding the floor) is equal to the received FTO (which is received by the talker taking the floor) plus the amount of delay on the channel of talker B’s microphone. Due to this observation, the system is free to manipulate the delay as long as talker B is not speaking. The most consistent way to vary the delay then is to alter the delay on talker B’s microphone during talker A’s turns. This approach leads to an implementation where the delay is varied at every other floor transfer.

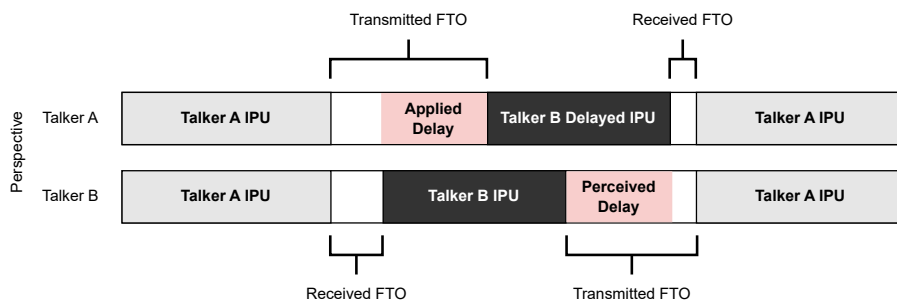


Figure 4.2: Demonstration that applying to delay to only one signal leads to a perceived delay by both talkers.

It should be noted that the effects of delay on conversation have been studied before. Previous studies focus on how delay impacts communication as a whole, and were conducted within the context of acceptable transmission line delays when designing telecommunication systems, originally with long-distance telephone lines (Brady, 1971), and more recently in digital communication systems, where delay has been studied along with the effects of packet loss (Michael and Möller, 2020). The work presented here is different in that the level of delay is varied within a range chosen based on human behavior, and also analyzes the resulting data from the modern perspective of conversational behavior, rather than measuring the direct impact on communication, or perceived quality of the conversations.

We hypothesize that talkers will adapt their speech in response to the perceived increase in magnitude and variability of their partner’s FTOs by increasing their own FTOs and

increasing the lengths of their IPU's and turns.

4.2 Methods

4.2.1 Participants

Nine pairs of young undergraduate participants were recruited as friends and screened for normal hearing (<20 dB HL) in the frequency range of speech (250 - 6000 Hz) using an Interacoustics AD226 diagnostic audiometer. Participants self-identified as native English speakers with no history of speech, language, or hearing disorders. All participants provided informed consent and the experiment was approved by the University of Waterloo Research Ethics Committee as application No. 45583. Data was collected by the author at the University of Waterloo and participants were remunerated as a thanks for their time.

4.2.2 Setup and Equipment

Participants were seated at desks in adjacent office rooms with more than 30 dB of transmission loss between them. Each participant had a headset microphone (DPA 4088) and a pair of headphones (Sennheiser HD 280 Pro). One microphone from a matched pair of iSEMCon EMX-7150 measurement microphones was placed in each room, approximately 1 meter away from the participants seating positions, and calibrated with a 94 dB SPL 1kHz test-tone. The headphones output gains were calibrated to a known dB SPL output level using a GRAS 45CA headphone test fixture and the headset microphone gains were set for each participant such that their partner would hear them at the same sound pressure level as if they were seated 1 meter away. The operator was seated away from the participant in one of the office rooms, and had a desktop microphone to communicate with participants between conversations. The experimental layout can be seen in Figure 4.3.

4.2.3 Task and Conditions

To elicit conversation among participants, all 12 spot-the-difference tasks from the DiapixUK corpus were performed (Baker and Hazan, 2011). Conversations took place in conditions that were combinations of quiet or noise and no delay or delay, with 3 replicates of each condition.

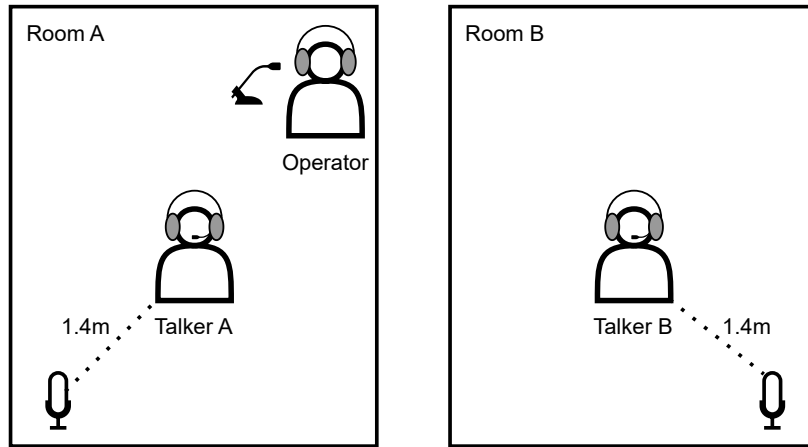


Figure 4.3: Experimental setup for the delay experiment. Participants were seated in adjacent office rooms, with the operator seated away from the participant in one of the rooms.

The noise used was a 10 minute 70 dB SPL babble noise, synthesized to have the same long-term spectrum as a 1 min segment of multi-talker speech babble (Bilger et al., 1984). The code used to conduct the sound texture synthesis was based on the study conducted by McWalter and McDermott (2018) and available through a git repository¹. Briefly, the sound texture synthesis algorithm involves two steps. In the first step, the algorithm takes an input signal and calculates a wide range of statistical estimates based on a model of peripheral auditory processing. In the second step, these statistics are imposed on a Gaussian white noise using an iterative process. The end result is an output signal of arbitrary length that has the same statistical properties as the input signal.

Delay was randomly sampled from a uniform distribution with bounds of 0 to 750 ms and varied at every other floor transfer of the conversation.

4.2.4 Experimental Procedure

The experiment was conducted in two visits. In the first, participants were provided and signed an information and consent letter. Following this, a general audiometric screening was conducted to ensure that all participants met the previously specified thresholds for normal-hearing. The first visit typically took 15 minutes to complete.

¹The code used for sound texture synthesis is available at: <https://github.com/rmcwalter/STSstep>

In the second visit, participants arrived to the testing room and were informed of the experimental setup and the goals of the task. Participants were provided information on completion of the task, including the maximum number of differences in each image (12) and the 6 minute time limit. There was no practice round of the task. Participants were further told that they may hear acoustic effects in their headphones, such as noise, but were not told that there will be delay present on the communication line. The second session was held in 3 blocks of 4 conversations and the participants were given a break after each block to prevent cognitive fatigue. Condition and image order were randomized for each set of participants, with a balanced weighting of conditions and Diapix scene types in each block. Each conversation ended either after 6 minutes or all 12 differences were found, whichever was first. Within blocks, after a conversation ended participants were provided the next set of images and the following conversation began shortly thereafter. Between blocks, participants were given a break that lasted at least 5 minutes. If participants took their headphones and microphone off during the break, the next block started with a recalibration of headset microphone levels.

After all 12 conversations had been completed, participants were provided with a secondary consent form, which asked for consent to publicize the data collected from the experiment. Permission was requested to publicize fully anonymized data from the conversation, such as VAD signals, for the purpose of open science. Permission was also requested to publicize the recorded speech signals from the conversation. Both were optional.

4.2.5 The Delay System

To vary the delay during communication, a near real-time state machine was developed to track floor transfers in conversation as they happened. The headset microphone signals from each talker were monitored block-by-block using a buffer size of 128 samples at 48kHz. Voice activity detection (VAD) was performed on each signal for each block by assessing whether the RMS value of the samples in the block was above an energy threshold set based on the background noise in the room. A time history of the VAD and speech signals were stored in separate buffers. The VAD history was used to assess recent speaking activity and detect the occurrence of turn-taking, and the speech history was used to enable the output of a delayed version of one talker’s speech. In post-hoc conversational state classification, one would bridge over short acoustic bursts and pauses based on the minimum pause and IPU lengths discussed in Section 4.1. However, performing an ongoing block-by-block correction for these minimums proved computationally intensive given the latency constraints of the system. In an attempt to avoid influence from these short acoustic bursts and gaps in speech, the conditions required to determine that turn-taking

had occurred were slightly relaxed, requiring that for the talker taking the floor 80% of the blocks in the last 90 ms must have been indicated as speech by the VAD method, and for the talker ceding the floor 80% of the blocks in the last 180 ms must have been identified as not containing speech. It was also required that the participant who is speaking must not already have the floor, to avoid repeated detection of a single turn-taking event.

As previously demonstrated in Figure 4.2, implementing delay was simplified by the fact that it is only necessary for delay to be added to one channel for it to be perceived by both talkers. The delay was always added to the microphone of talker B, to avoid any possibility of a participant receiving both a delayed version of the signal through the system, and a non-delayed version of the signal through the operator’s microphone. Although this was not intended, it could happen, for example, when informing the participants that the time limit had been reached. Delay was implemented in a two-step process. First, when the floor transferred from talker A to talker B, a new random delay value was drawn. Second, when the floor transferred from talker B to talker A, the delay on talker B’s microphone was manipulated. Delay was tracked and manipulated using a pair of pointers. The first pointer tracked the current (real-time) position of speech, and the second tracked the delayed position of speech. The pointers were incremented with each incoming block of audio, such that the current pointer always corresponded to the most recent block of audio. The audio block output to the headphones of talker B was always the block immediately preceding the current pointer. Given that the current pointer was always the most recent block of audio, this signal could just be passed through the system directly. Likewise, the signal output to the headphones of talker A was always the block just before the delayed pointer, which could be accessed via the time history of speech.

At the start of a conversation, the current and delayed pointers were identical, thus there was no delay. However, once a new delay value was randomly drawn, it needed to be implemented. The first manipulation was always to add delay, given that there was no initial delay on the line. To add delay, the corresponding pointer was withheld from advancing with the current pointer (i.e., it was held constant despite the current pointer still incrementing block-by-block) until the difference between the current and delayed pointer was equal to the drawn amount of delay. To remove delay, the change in delay was calculated as the difference between the current delay and the newly drawn delay value, and the processing algorithm waited until talker B had been silent for an equal amount of time, upon which the delayed pointer was skipped forward such that the current delay matched the randomly drawn value. Since talker B was silent for the entire duration that the pointer was skipped forward, it was guaranteed that no speech would be lost.

Piloting by the researchers revealed that there was no audible artifacts when manipulating delay using this method.

It is worth noting that the method for voice-activity detection in near real-time will be less accurate than a post-hoc non-causal method such as that used in the following statistical analysis of the timing of turn-taking. However, in all cases delays were added at a substantial amount of the floor transfers, as will be discussed along with the results.

4.2.6 Data Postprocessing

To attenuate low frequency HVAC noise in the testing rooms, recordings from headset microphones were first zero phase filtered with a cutoff frequency of 100 Hz.

Voice activity detection was then performed on the filtered signals by first computing the power, in dBFS, of 5 ms windows with 1 ms of overlap. A power threshold was set 30 dB down from the 99th percentile of the power distribution for each talker, in each conversation. Any window with a power greater than this threshold was classified as containing speech. Silent segments less than 180 ms were classified as gaps within speech and bridged over to avoid misidentifying a stop-gap as a pause, as described by Heldner and Edlund (2010). Sound bursts less than 90 ms were thought to be nonspeech events, such as tapping or coughing, and similarly bridged over. Speech levels in each of these windows were computed by adding a constant set during the calibration process to the power.

To assess differences in speaking rate across conditions, syllables were identified using the Syllable Nuclei v3 algorithm from De Jong et al. (2021), with parameters chosen such that the voice activity detection would happen in the same manner as described previously.

The speech activity signals were run through a conversational state classification algorithm, which identified IPUs, FTOs, and pauses from the pair of voice activity signals in a conversation. Additional features within each IPU and FTO were calculated, such as duration, average level of speech, and articulation rate. To ensure that analysis of the FTOs was performed relative to the talker taking the floor, the classification algorithm was run twice. First, both real-time headset microphone signals were input, and the IPUs, FTOs, and pauses corresponding to the talker whose microphone was being delayed were kept. For the other talker, their real-time signal along with the delayed signal of the other talker were input, and the IPUs, FTOs, and pauses for the talker whose microphone was not being delayed were kept.

4.2.7 Statistical Methods

Analysis across conditions typically used generalized (GLMM) or linear mixed-effects models (LMM) with the interaction between delay and background noise as a fixed effect, and the talker (or pair) and replicate as random effects. Models were fit using functions from the lme4 package in R (*glmer* for GLMM, *lmer* for LMM). In some models, other fixed effects were included and will be discussed along with the results. Marginal means were estimated from the fit models using the emmeans package. For GLMMs results are interpreted from the model coefficients, and confidence intervals and p-values were computed with the lmerTest package using a Wald t-distribution approximation. For LMMs, a type III analysis of variance was performed using Satterthwaite’s method, and results interpreted from the ANOVA.

4.3 Results

4.3.1 Delay implementation verification

First, to verify the effectiveness of the delay implementation algorithm, the distributions of transmitted and received FTOs in the conversations with delay were compared, and clear differences are observed, as shown in Figure 4.4a. Also evaluated was the mean delay in each conversation along with the ratio of the number of instances the delay was manipulated to the number of eligible floor-transfers found by the post-processing turn-taking analysis, seen in Figures 4.4b and, 4.4c, respectively. Although some conversations had their delay changed very infrequently, there was still a considerable amount of delay present on average in all conversations (> 250 ms) therefore none were excluded from analysis.

4.3.2 Floor-transfer offsets

The FTO distributions for each condition were modeled using the `geom_density` function in R, and are displayed in Figure 4.5a. The distributions are almost completely on top of each other, although the peak does appear to be slightly lower in noise, implying a broadening of the distribution.

The FTO distribution was shifted and truncated such that it only included positive values and modeled as a gamma distribution. To determine the shift that would result in

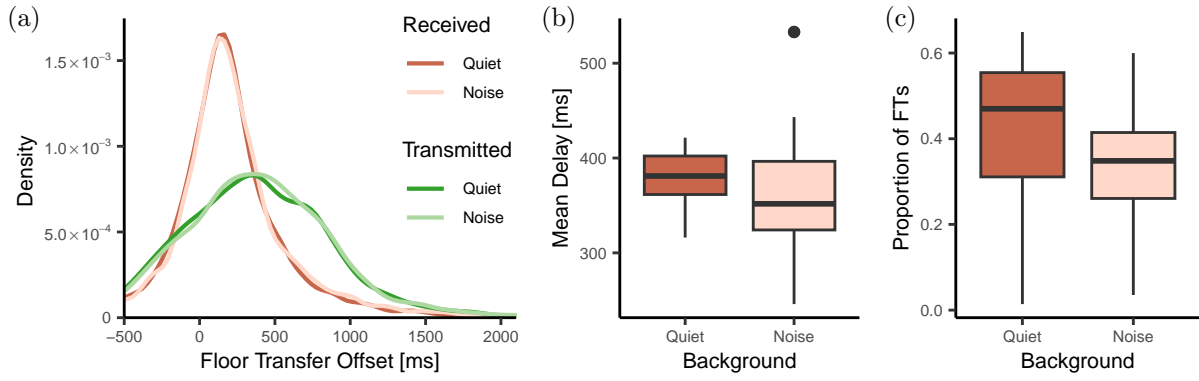


Figure 4.4: Distributions of the received and transmitted FTOs in quiet and noise, in the conversations where delay was present (a). Also shown are boxplots of (b) the mean delay implemented in quiet vs. noise and (c) the proportion of eligible floor transfers at which the delay was manipulated.

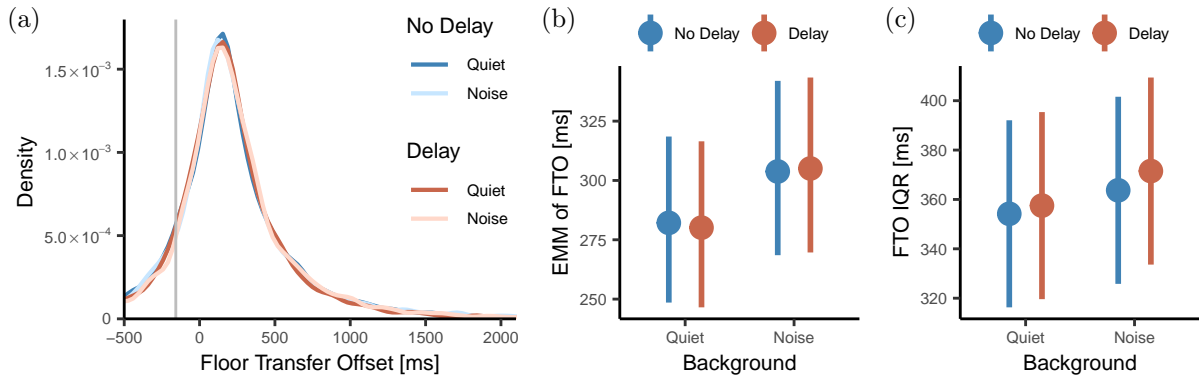


Figure 4.5: Distribution of floor-transfer offsets by condition (a), and the estimated marginal means of (b) FTO duration and (c) interquartile ranges of the FTO. The results are grouped by background condition, and color indicates the presence of delay.

Fixed Effect	β -value	95% C.I.	t-value	Pr(> z)	
Intercept	6.09	[6.01, 6.17]	150.19	< .001	***
Noise	0.05	[0.01, 0.08]	2.76	< .005	**
Delay	-0.01	[-0.04, 0.03]	-0.26	> .05	
Pre F.T. Syll.	-0.09	[-0.10, -0.08]	-13.99	< .001	***
Post F.T. Syll.	0.08	[0.06, 0.09]	10.92	< .001	***
Pre F.T. A.R.	0.02	[0.01, 0.03]	3.24	< .01	**
Post F.T. A.R.	-0.08	[-0.10, -0.07]	-12.86	< .001	***
Noise:Delay	0.01	[-0.04, 0.06]	0.30	> .05	

Table 4.1: Statistical results for the GLMM fit to the floor transfer offset distribution.

the best fit, the variance and skewness of all FTOs were calculated and used to estimate the mean of the gamma distribution with the same variance and skewness. FTOs that were less than the difference between the estimated gamma mean and the empirical mean of all FTOs were excluded (< -157 ms, indicated by the vertical bar in Figure 4.5a). A generalized linear mixed-effects model of the form: $\text{FTO} \sim \text{Noise} * \text{Delay} + \text{Pre FT Syllable Count} + \text{Post FT Syllable Count} + \text{Pre FT Articulation Rate} + \text{Post FT Articulation Rate} + (1 | \text{Talker})$ was fit to a conditional gamma distribution with a log link function. The syllable count and articulation rate in the IPUs directly around the floor transfers were also included in the model as fixed effects, as the FTO has been shown to depend on these characteristics (Roberts et al., 2015).

The results of this model revealed a significant increase of the FTO in noise [$t(14297) = 2.75, p < 0.01$], but not in delay. The model results also revealed, as expected, significant effects of the syllable counts and articulation rates in the IPUs both before and after the floor transfer. Statistical results are summarized in Table 4.1. The estimated marginal means were extracted from the GLMM and corrected for the shift discussed earlier and are displayed in Figure 4.5b.

The variability of the FTO was measured using the interquartile (IQR) ranges of the FTOs in each conversation. A linear mixed effects model of the form $\text{IQR} \sim \text{Noise} * \text{Delay} + (1 | \text{Talker})$ was fit. The replicate was excluded as a random effect as it resulted in a singular fit of the model. An analysis of variance on the LMM revealed no significant effects of noise [$F(1, 195) = 1.15, p = 0.29, \eta_p^2 = 0.006$], delay [$F(1, 195) = 0.26, p = 0.61, \eta_p^2 = 0.001$], or the interaction [$F(1, 195) = 0.04, p = 0.84, \eta_p^2 = 0.0002$] on the IQR of the FTO.

4.3.3 Interpausal units

The interpausal unit duration is a measure of how long people talk each time they talk. IPU duration distributions were estimated by condition using the `geom_density` function, and are displayed in Figure 4.6a. The plots of distributions suggest that in noise, the IPU distribution shifts slightly to the right.

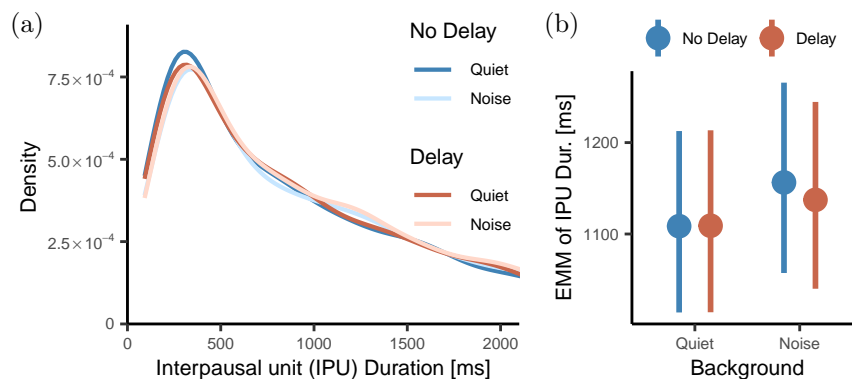


Figure 4.6: Distributions of interpausal unit duration by condition (a) and (b) estimated marginal means of IPU duration. The results are grouped by background condition, and color indicates the presence of delay.

The effect of condition on IPU duration was modeled using a generalized linear mixed effects model of the form $\text{IPU Durations} \sim \text{Noise} * \text{Delay} + (1 | \text{Talker})$ with the same family and link function as was used for the FTO analysis. The inclusion of replicate as a random effect resulted in a singular model, so it was excluded. Before fitting, the IPUs were shifted by the minimum possible duration of 90 ms so that the distribution starts at 0.

The results of this model, summarized in Table 4.2, revealed a significant positive effect of noise on the IPU [$t(22528) = 2.61, p < 0.01$], indicating that IPUs become longer when conversing in noise, but no effect of delay. Figure 4.6b displays the estimated marginal means of IPU duration by condition.

4.3.4 Pauses

The duration and rate of pauses can also be indicators of conversational difficulty. The distributions of the durations of pauses by condition were estimated and are shown in Figure 4.7a. The distributions appear to broaden and shift right in both noise and delay.

Fixed Effect	β -value	95% C.I.	t-value	$\Pr(> z)$	
Intercept	6.93	[6.83, 7.02]	139.61	< .001	***
Noise	0.05	[0.01, 0.08]	0.61	< .01	**
Delay	0.00	[-0.03, 0.04]	0.03	> .05	
Noise:Delay	-0.19	[-0.07, 0.03]	-0.74	> .05	

Table 4.2: Statistical results for the GLMM fit to the IPU duration distributions.

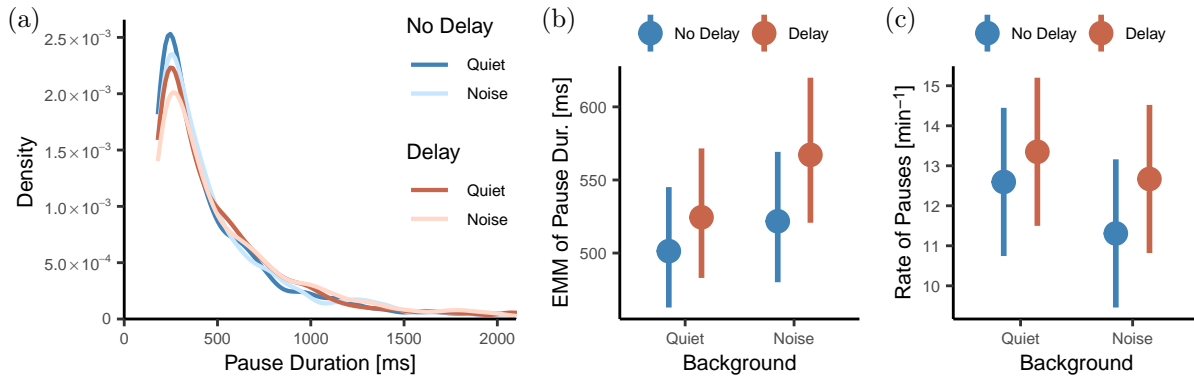


Figure 4.7: The distributions of pause duration by condition (a), and the estimated marginal means of (b) pause duration (ms), and (c) rate of pauses (pauses per minute of speaking time). The results are grouped by background condition, and color indicates the presence of delay.

A generalized linear mixed model of the form: Pause Duration \sim Noise*Delay + Pre Pause IPU Duration + Post Pause IPU Duration + (1 | Talker) + (1 | Replicate) was fit to a gamma distribution with a log link function. Before fitting, the pause durations were shifted by the minimum possible duration of 180 ms, defined by the bridging that occurred during voice activity detection. The durations of the IPU immediately surrounding the pauses were included as fixed effects in the model, to account for the significant increase in IPU duration in noise, as described in 4.3.3.

The results from the GLMM are summarized in Table 4.3. A significant negative effect of the duration of the IPU before a pause was found [$t(6101) = -9.21, p < 0.001$], and a borderline significant effect of delay was found [$t(6101) = 1.76, p = 0.08$], but not for any of the other fixed effects. The estimated marginal means of pause duration are shown in Figure 4.7b.

Fixed Effect	β -value	95% C.I.	t-value	Pr(> z)	
Intercept	5.77	[5.65, 5.90]	88.85	< .001	***
Noise	0.06	[-0.02, 0.14]	1.51	> .05	
Delay	0.07	[-0.01, 0.15]	1.76	< .1	.
Pre Pause IPU Dur.	-0.12	[-0.15, -0.10]	-9.21	< .001	***
Post Pause IPU Dur.	0.01	[-0.02, 0.04]	0.77	> .05	
Noise:Delay	0.10	[-0.06, 0.17]	0.97	> .05	

Table 4.3: Statistical results for the GLMM fit to the durations of pauses.

The rate of pauses in conversation was also analyzed. The rate was calculated by dividing the number of pauses by each talker by the sum of the IPU and pause durations of that talker, therefore no correction for delay was necessary. An LMM of the form: Pause Rate \sim Noise*Delay + (1 | Talker) was fit to the rates of pauses by each talker. Replicate was excluded as a random effect as it resulted in a singular fit of the model. An analysis of variance on the LMM revealed significant effects of noise [$F(1, 195) = 9.13, p < 0.01, \eta_p^2 = 0.04$] and delay [$F(1, 195) = 10.55, p < 0.01, \eta_p^2 = 0.05$] but not of the interaction [$F(1, 195) = 0.86, p = 0.35, \eta_p^2 = 0.004$]. The estimated marginal mean pause rates are plotted in Figure 4.7c.

4.3.5 Overlaps-within

Overlaps-within (OW) are a subset of IPU that exist completely within a partner’s turn. The distributions of OW duration were estimated and are plotted in Figure 4.8a. A GLMM

Fixed Effect	β -value	95% C.I.	t-value	$\Pr(> z)$	
Intercept	5.48	[5.37, 5.58]	101.88	< .001	***
Noise	-0.83	[-0.19, 0.02]	-1.53	> .05	
Delay	0.14	[0.04, 0.24]	2.82	< .01	**
Noise:Delay	-0.11	[-0.26, 0.03]	-1.52	> .05	

Table 4.4: Statistical results for the GLMM fit to the duration of overlaps-within.

of the form: OW Duration \sim Noise*Delay + (1 | Talker) + (1 | Replicate) was fit to the durations of overlaps-within. The results, displayed in Table 4.4, reveal a significant positive effect of delay on the duration of overlaps-within [$t(2693) = 2.81, p < 0.01$]. The estimated marginal means are plotted in Figure 4.8b.

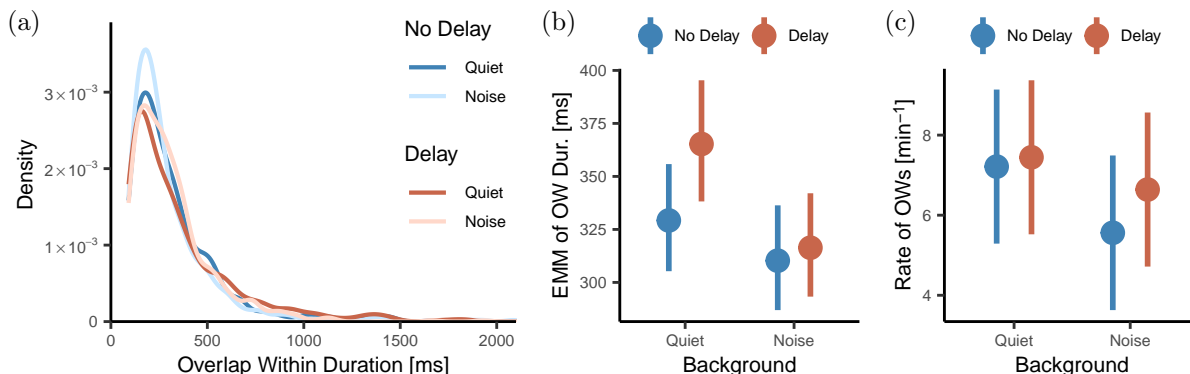


Figure 4.8: Distributions of overlaps-within duration by condition (a), and the estimated marginal means of (b) OW duration (ms), and (c) rate of OWs (OWs per minute of speaking time). The results are grouped by background condition, and color indicates the presence of delay.

The rate of overlaps-within was computed as the count of overlaps within divided by the total IPU time (including OWs) for each talker in each conversation. An LMM of the form: OW Rate \sim Noise*Delay + (1 | Talker) + (1 | Replicate) fit to the OW rates. An analysis of variance revealed a significant effect of noise [$F(1, 194.02) = 4.06, p < 0.05, \eta_p^2 = 0.02$], but not delay [$F(1, 194.02) = 1.16, p = 0.28, \eta_p^2 = 0.006$] or the interaction [$F(1, 194.02) = 0.48, p = 0.49, \eta_p^2 = 0.003$]. The estimated marginal mean rates of overlaps-within is plotted in 4.8c.

4.3.6 Turn duration and turn-taking rate

Turn durations were analyzed in the same manner as the IPU durations. The distributions are shown in 4.9a. A GLMM of the form: Turn Duration \sim Noise*Delay + (1 | Talker) was fit. Statistical results of the GLMM fit to the distributions of turn durations is included in Table 4.5. A borderline significant positive effect of noise on turn duration was found ($p = 0.0525$). The estimated marginal means of turn duration by condition are plotted in 4.9b.

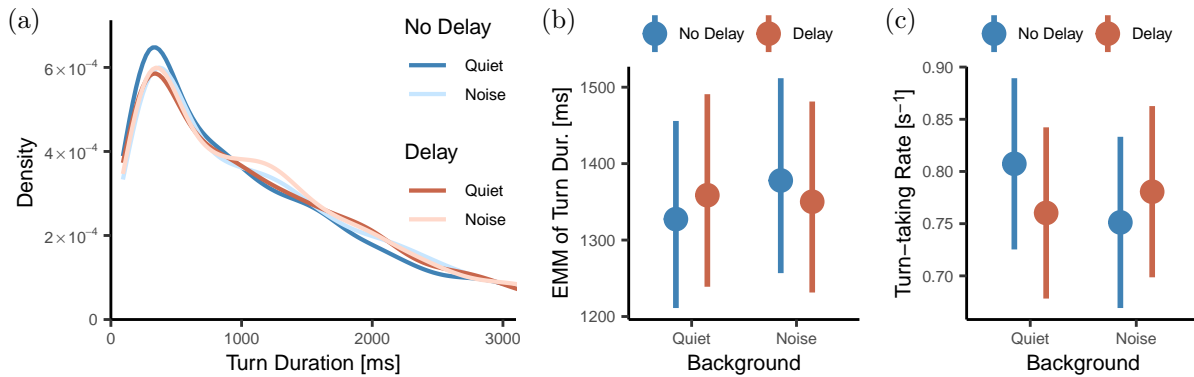


Figure 4.9: Estimated marginal means of (a) turn duration (ms) and (b) turn-taking rate (turns per second). The results are grouped by background condition, and color indicates the presence of delay.

Fixed Effect	β -value	95% C.I.	t-value	Pr($> z $)	
Intercept	7.12	[7.02, 7.21]	141.37	$< .001$	***
Noise	0.04	[0.00, 0.08]	1.94	$< .1$.
Delay	0.03	[-0.02, 0.07]	1.19	$> .05$	
Noise:Delay	-0.05	[-0.11, 0.01]	-1.58	$> .05$	

Table 4.5: Statistical results for the GLMM fit to the distribution of turn durations.

The turn-taking rate was computed for each conversation as the inverse of the mean turn duration. This method was chosen as it avoids accounting for the effect delay has on the duration of the conversation. An LMM of the form: Turn-taking Rate \sim Noise*Delay + (1 | Talker) was fit. An analysis of variance revealed no significant effect of noise [$F(1, 195) = 0.64, p = 0.42, \eta_p^2 = 0.003$] or delay [$F(1, 195) = 0.16, p = 0.69, \eta_p^2 = 0.0008$], but a borderline significant effect of the interaction [$F(1, 195) = 2.94, p = 0.09, \eta_p^2 = 0.01$]. The estimated marginal mean turn-taking rates are plotted in 4.9c.

4.3.7 Speaking and listening time

The proportion of speaking time is defined here as the ratio of the speaking time of the dominant talker to the total speaking time in each conversation. The dominant talker was determined as the interlocutor in each pair that talked the most across all conversations. A linear mixed model of the form: $\text{Speaking Time} \sim \text{Noise*Delay} + (1 \mid \text{Talker})$ was fit to the proportions of speaking time. An analysis of variance revealed a significant effect of delay [$F(1, 96) = 7.69, p < 0.01, \eta_p^2 = 0.07$], a borderline significant effect of the interaction [$F(1, 96) = 3.24, p = 0.08, \eta_p^2 = 0.03$], and no effect of noise [$F(1, 96) = 0.43, p = 0.51, \eta_p^2 = 0.005$]. The estimated marginal mean speaking times are plotted in Figure 4.10a.

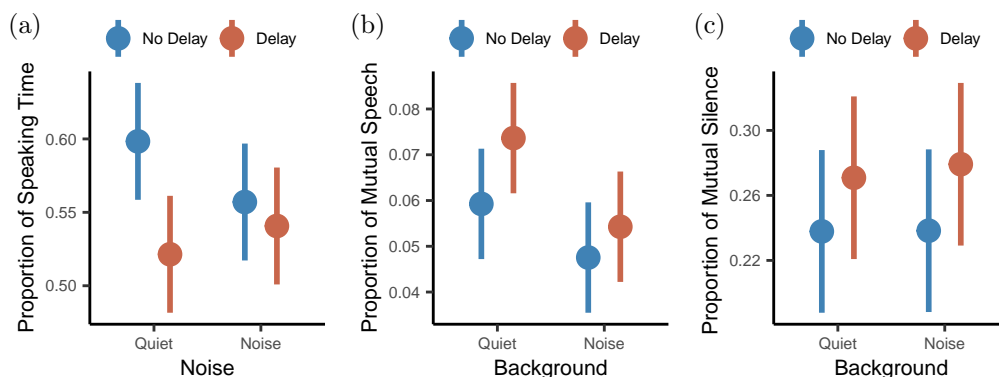


Figure 4.10: Estimated marginal means of (a) speaking time by the dominant talker, (b) overlapped speaking time, and (c) overlapped silence time. The results are grouped by background condition, and color indicates the presence of delay.

To assess the impact on communication generally, the overlapped portions of the conversation were looked at. We define mutual talking time as the proportion of the conversation spent with both interlocutors speaking and mutual silence time as the proportion of the conversation spent with neither speaking. In both cases, the real-time microphone signals from both talkers were used, although a post-hoc analysis revealed no significant difference between the results obtained here and the results that would be obtained if the delayed microphone signal were used instead.

An LMM of the form $(\text{Mutual Talking or Mutual Silence Time}) \sim \text{Noise*Delay} + (1 \mid \text{Talker}) + (1 \mid \text{Replicate})$ was fit to the mutual talking and mutual silence time. Analyses of variance revealed the following effects. Mutual talking: significant effects of noise [$F(1, 202) = 45.2, p < 0.001, \eta_p^2 = 0.18$] and delay [$F(1, 202) = 20.84, p <$

0.001, $\eta_p^2 = 0.09$], and a borderline significant effect of the interaction [$F(1, 202) = 2.73, p = 0.099, \eta_p^2 = 0.01$]. Mutual silence: a significant effect of delay [$F(1, 204) = 103.43, p < 0.001, \eta_p^2 = 0.34$], but not noise [$F(1, 204) = 1.45, p = 0.23, \eta_p^2 = 0.007$] or the interaction [$F(1, 204) = 1.17, p = 0.28, \eta_p^2 = 0.006$]. The estimated marginal mean mutual talking and mutual silence times are displayed in Figures 4.10b and 4.10c, respectively.

4.3.8 Speech acoustics and articulation rate

Characteristics of the talkers' speech were also analyzed. An LMM was fit to the mean speech level (in dB SPL) in each conversation. The model was of the form: Level \sim Noise*Delay + (1 | Talker). An analysis of variance revealed a significant effect of noise [$F(1, 195) = 554.95, p < 0.001, \eta_p^2 = 0.74$], but no effect of delay [$F(1, 195) = 0.005, p = 0.94, \eta_p^2 = 0.00003$] or the interaction [$F(1, 195) = 1.67, p = 0.20, \eta_p^2 = 0.009$]. The estimated marginal mean speech levels, in dB SPL, are plotted in Figure 4.11a.

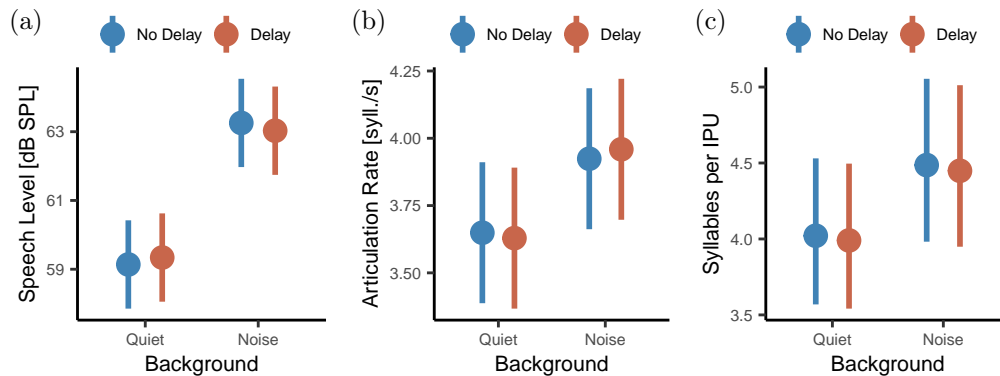


Figure 4.11: Estimated marginal means of (a) speech level (dB SPL), (b) articulation rate (syllables per second), and (c) number of syllables per IPU. The results are grouped by background condition, and color indicates the presence of delay.

The mean articulation rate over the course of the conversation was computed as the number of syllables spoken by each talker divided by their total speaking time. An LMM of the form: Articulation Rate \sim Noise*Delay + (1 | Talker) + (1 | Replicate) was fit to the data. An analysis of variance revealed a significant effect of noise [$F(1, 193) = 43.28, p < 0.001, \eta_p^2 = 0.18$], but not delay [$F(1, 193) = 0.03, p = 0.87, \eta_p^2 = 0.0001$] or the interaction [$F(1, 193) = 0.36, p = 0.55, \eta_p^2 = 0.002$]. The estimated marginal mean articulation rates are plotted in Figure 4.11b.

Fixed Effect	β -value	95% C.I.	z-value	Pr(> z)	
Intercept	1.39	[1.27, 1.51]	22.87	< .001	***
Noise	0.11	[0.09, 0.13]	12.18	< .001	***
Delay	-0.01	[-0.03, 0.01]	-0.83	> .05	
Noise:Delay	-0.00	[-0.03, 0.02]	-0.50	> .05	

Table 4.6: Statistical results for the GLMM fit to number of syllables per IPU.

Although the mean articulation rate increases in noise, we can also analyze if the number of syllables in each IPU is increasing. A GLMM of the Poisson family was fit using a log link function to the data. The model had the form: Number of Syllables \sim Noise*Delay + (1 | Talker) + (1 | Replicate). A significant effect of noise was observed ($p < 0.001$), but no significant effects of delay or the interaction were found. The estimated marginal mean number of syllables per IPU are plotted in [4.11c](#).

4.4 Discussion

4.4.1 Effects of delay

The results presented suggest that the increased magnitude and variability of FTOs are not used by talkers to infer when their partner(s) may be experiencing difficulty. The duration of neither IPU nor FTOs increased in response to a simulated increase in FTO duration and variability, revealing that there was no effect of delay on turn-taking behavior. Given that talkers have been shown to adapt their behavior in experiments where increased difficulty is only present for one talker (Beechey et al., 2020; Hazan and Baker, 2011), this result is unexpected and implies that the timing of turn-taking is not used as an indicator of difficulty by the partner. Otherwise a behavioral adaptation would have been exhibited in response, as was observed in the above studies. However, it is possible that the task interfered with this result as a longer FTO could simply be attributed to a demand of the task, such as a visual search being performed. For this reason, it may be appropriate to evaluate the effects of delay in a free form conversation setting, where talkers would be more naturally communicating and therefore unable to attribute the timing differences to a component of the task at hand. Another possible explanation for the lack of effect of delay is due to the format of the conversations. Given that modern people are quite accustomed to communicating through systems that have substantial latency present (e.g., hundreds of milliseconds over Zoom (Boland et al., 2022)), it's possible that delay could

be subconsciously attributed to the medium of the conversation, given that conversing via headphones and microphones may more closely resemble a telephone call than a face-to-face conversation. Although the lack of effects of delay contradicts our expectations, it could explain why in previous studies, hearing-impaired talkers have been shown to exhibit a larger increase in FTO in noise compared to their normal-hearing interlocutors (Petersen et al., 2022). If talkers primarily adapt their FTO in response to their own difficulty, then there must be other cues used as adaptations to an interlocutor’s difficulty.

However, delay did affect communication. Similar to the results found in Brady (1971), we found that a higher proportion of conversations taking place in delay consisted of both overlapped speech and overlapped silence. One interpretation of this is that there was more interruptions or confusions during the conversations with delay. This possibility is corroborated by the significant increase of the duration of overlaps-within in delay. There was also a borderline significant increase in pause duration in delay, along with a significant increase in the rate of pauses. This could again support the previous suggestion, as talkers may end their turn expecting their partner to take the floor, and then begin again when they have not heard their partner respond yet. In reality, the partner may have begun talking, but the delayed signal hasn’t been received yet.

4.4.2 Effects of noise

The effects of noise did not agree with expectations surrounding the impact of increased difficulty on conversational dynamics. Several previous studies have found that the duration of FTOs, the variability thereof, and the duration of IPU increase in noise (Sørensen et al., 2021; Petersen et al., 2022; Sørensen et al., Submitted). However, despite using a 70 dB SPL noise, we observed only a small increase in FTO and IPU durations, with no increase in the variability of the FTO. Further, the estimated marginal mean speech levels in noise were centered around 63 dB SPL, which would indicate a -7 dB signal-to-noise ratio of speech. These findings suggest that the noise used was not as effective a masker of speech as expected. An analysis of the spectral content of the speech-shaped babble noise used revealed that the perceptually weighted level of the noise was ~ 65 dBA. The results, when interpreted with this level in mind, are more aligned with past studies. Additionally, speech levels could have been higher than normal, as participants wore closed-back (occluded) headphones, which reduced the air-conducted auditory feedback of their own voice, potentially increasing the speech levels they produced. Thus, the SNR received in the noise condition would have been higher than if talkers own speech was heard at natural level without attenuation from the occluding headphones, further decreasing the difficulty induced by the noise.

For these reasons, we suggest that the noise used here did not introduce as much difficulty as expected, and therefore the intended difficult control condition was not present. Despite this, we expect that the effects of delay are still valid and interpretable. The lack of effect of delay on the typically analyzed turn-taking dynamics suggest that other cues, aside from timing, must be used to infer difficulty in a partner. These cues could include acoustic or behavioral adaptations, adjustments in the content or syntactic structure of language used, or even direct requests to speak louder or repeat phrases. One possible explanation is that the timing of turn-taking is not used in isolation to infer difficulty, but that it may be used in conjunction with other cues. Given that the conversations taking place in noise were not as difficult as expected, the interaction effects could have been masked. For this reason, a retest of the experiment is suggested with a noise level that will introduce a significant amount of difficulty, such as a 70 dBA noise rather than 70 dB SPL. One example is the multitalker babble from the ICRA noises, which were designed to mimic both spectral and temporal properties of speech (Dreschler et al., 2001).

4.5 Conclusions

This study evaluated the effect of simulated increases in duration and variability of floor-transfer offsets in the form of a randomly varying delay to assess if talkers use the timing of their partner’s turn-taking to infer that difficulty is being experienced. Contrary to our hypothesis, no effect of delay was found on turn-taking dynamics, suggesting that talkers must be using cues that aren’t related to the timing of turn-taking to adapt their speech in response to a conversational partner’s difficulty. However, due to a suspected lack of masking ability by the background noise used, a retest is proposed to ensure that there is a control condition that is introducing difficulty into the conversation.

Acknowledgments

This work was funded by the Natural Sciences and Engineering Research Council of Canada (RGPIN-2021-03085).

Chapter 5

Conclusions

In this thesis, two approaches for understanding talker sensitivity to turn-taking were explored. These studies were targeted at better understanding listening effort in conversation, and aim to work towards ecologically valid assessments of hearing loss, where listening effort and the effects of hearing impairment can be measured naturalistically in the environment in which the most detrimental effects are experienced. The first approach, which utilized physiological measures of cognitive effort to assess divided attention in conversation, was the first study to analyze pupil response at a fine enough temporal scale to reveal how attention changes happen around turn-taking. This approach revealed systematic pupil responses that exist across people and environmental conditions, which seemed to correspond with expectations of the reallocation of effort and attention. The second approach utilized a new experimental design, in which talkers perception of their conversational partner's behavior was altered using real-time manipulations of delay on the communication line between talkers. This study artificially increased the floor transfer offsets perceived by talkers in conversation, mimicking what has been observed in conversations that take place in difficult acoustic environments. However, despite the expectation that talker's would perceive that their partner was experiencing difficulty due to the increased FTO, no adaptations in behavior were observed, suggesting that talkers must use other cues to determine a partner's difficulty in conversation. Additionally, a custom audio processing framework was developed to facilitate the behavioral experiment performed in the second approach. This framework was then modularized to be adaptable to numerous other auditory feedback and conversational applications, increasing accessibility to experimental designs where researchers can manipulate the acoustics and timing dynamics of speech in near real-time.

5.1 Pupil response to turn-taking

The method introduced in Chapter 2 shows promising results, revealing significant and systematic pupil responses around turn-taking that also varied based on conversational condition and other factors such as the duration of a talkers turn. To our knowledge, this was the first to study to analyze pupil response within conversations, beyond analyzing how pupil size differed between conversational conditions and overall speaking and listening time as in Li et al. (2020); Aliakbaryhosseinabadi et al. (2023). We also believe this is the first study to apply the temporal response function approach to estimate pupil response to stimuli, as previous studies tend to measure the pupil response using trial or epoch based approaches or other time-series based approaches. Although this analysis was performed on a general conversational dataset that was previously collected, future work could include purpose-designed experiments to extract some useful information from the pupillometry. One example could be a task-based conversational study that recruits groups of three participants, with two participants conversing at a time and the third following along. This would enable separation of the cognitive effort related to speaking, and isolate the response only to listening and task demands. By comparing results between completing the task and following along, we could infer how listening effort interacts with speaking effort and how this interaction changes as a result of conversational difficulty. Additionally, by purpose designing experiments we can also take more precautions to limit interference of other factors on the pupil response, such as having the participants seated in separate rooms, without view of each other, to minimize effects from looking between the task and the partner. Other options for future work could include comparing the pupillary responses of normal-hearing and hearing-impaired talkers using the method introduced, or applying the same method of analysis to other physiological signals, such as eye-gaze or neural signals. Eventually, a multimodal model would likely be most beneficial, as the effects that correlate strongly across many different types of response signals are likely to be related to the task demands rather than some other environmental factor that would only influence one of the signals, such as the pupil response to light exposure.

5.2 Real-time audio processing framework

An audio processing framework was successfully developed and presented in Chapter 3, and then implemented for a behavioral experiment, as described in Chapter 4. Some example modules were described to demonstrate that it is straightforward to build a processing module for inclusion in the system. The latency of the system was evaluated and found to

be acceptable for many auditory experiments, even on entry-level hardware. Future directions for this framework include adapting to a fully modular form, including UI elements. Currently, some of the code of the main framework needs to be modified to include custom modules. However, to be fully accessible and have minimal barrier-to-entry the system should be fully modular. In addition, documentation and tutorials should be developed to demonstrate how to build custom processing modules.

5.3 Delay in conversation

Although no significant effect of delay was found on the metrics of conversational dynamics, as presented in Chapter 4, this study should be redone with a proper noise signal. Despite the suspected lack of difficulty introduced by the noise, the effects of delay should still be valid, although the interaction between noise and delay may change when using a more difficult noise signal. The lack of effect of delay on conversational dynamics suggests that talkers must use other cues to identify that a conversational partner is experiencing difficulty. Some possible cues include an increase in the level of speech, a decrease in the rate of speech, or semantic indications, such as a request for the repetition of speech. Possible future directions of research include analyzing sensitivity of talkers to each of these cues from their conversational partners. Although some of these may be difficult to isolate, it is possible that the framework introduced in Chapter 3 could be used to either simulate or introduce difficulty for one talker only. One example of this is by attenuating a talker’s speech signal that is received by their partner in the presence of noise, which would not result in the same increased Lombard effect that playing a louder noise for one talker than the other would, but has the same effect of decreasing the signal-to-noise ratio. Another potential experiment to simulate the effects of increased difficulty to test cue sensitivity could be to artificially slow down talkers speech by using a pitch-corrected time dilation on one talkers channel, or by lengthening pauses, thereby increasing turn lengths.

5.4 Implications of the findings

Although the motivation of these studies is to better understand hearing loss and the effects thereof, a secondary benefit is that the results lead towards better design and verification of hearing aids. By deepening our understanding of ‘normal’ behavioral and physiological responses in conversation, we move towards the ability to use conversation as a naturalistic method to assess new hearing aids. As hearing assistive features become more

sophisticated, the typical approaches used for assessing listening effort, such as speech understanding in noise, end up insufficient, for example when evaluating dynamic features that are meant to vary based on the current action of the wearer. Therefore, this previously mentioned notion of ecological validity is not only an advantage, but rather a necessity for appropriate evaluation of effectiveness of said devices. Further, given the recent advances in wearable technologies, it may be ideal for hearing aids to incorporate physiological and behavioral responses to adaptively adjust the level of aid in real-time, or to monitor the cognitive health of wearers over extended periods of time. For this reason, understanding how physiology can be used as indicators of effort in conversation is important for broadening the usefulness of hearing aids beyond being simply listening assistive devices, and transforming them into health assistive devices.

References

- Aliakbaryhosseinabadi, S., Keidser, G., May, T., Dau, T., Wendt, D., and Rotger-Griful, S. (2023). The Effects of Noise and Simulated Conductive Hearing Loss on Physiological Response Measures During Interactive Conversations. *Journal of Speech, Language, and Hearing Research*, 66(10):4009–4024.
- Baker, R. and Hazan, V. (2011). DiapixUK: task materials for the elicitation of multiple spontaneous speech dialogs. *Behavior Research Methods*, 43(3):761–770.
- Beatty, J. (1982). Task-evoked pupillary responses, processing load, and the structure of processing resources. *Psychological Bulletin*, 91(2):276–292.
- Beechey, T., Buchholz, J. M., and Keidser, G. (2018). Measuring communication difficulty through effortful speech production during conversation. *Speech Communication*, 100:18–29.
- Beechey, T., Buchholz, J. M., and Keidser, G. (2020). Hearing Impairment Increases Communication Effort During Conversations in Noise. *Journal of Speech, Language, and Hearing Research*, 63(1):305–320.
- Bilger, R. C., Nuetzel, J. M., Rabinowitz, W. M., and Rzeczkowski, C. (1984). Standardization of a Test of Speech Perception in Noise. *Journal of Speech, Language, and Hearing Research*, 27(1):32–48.
- Boland, J. E., Fonseca, P., Mermelstein, I., and Williamson, M. (2022). Zoom disrupts the rhythm of conversation. *Journal of Experimental Psychology: General*, 151(6):1272–1282.
- Bradley, M. M., Miccoli, L., Escrig, M. A., and Lang, P. J. (2008). The pupil as a measure of emotional arousal and autonomic activation. *Psychophysiology*, 45(4):602–607.

- Brady, P. T. (1971). Effects of Transmission Delay on Conversational Behavior on Echo-Free Telephone Circuits. *Bell System Technical Journal*, 50(1):115–134.
- Brusco, P., Vidal, J., Beňuš, , and Gravano, A. (2020). A cross-linguistic analysis of the temporal dynamics of turn-taking cues using machine learning as a descriptive tool. *Speech Communication*, 125:24–40.
- Bögels, S. (2020). Neural correlates of turn-taking in the wild: Response planning starts early in free interviews. *Cognition*, 203:104347.
- Bögels, S., Casillas, M., and Levinson, S. C. (2018). Planning versus comprehension in turn-taking: Fast responders show reduced anticipatory processing of the question. *Neuropsychologia*, 109:295–310.
- Crosse, M. J., Di Liberto, G. M., Bednar, A., and Lalor, E. C. (2016). The Multivariate Temporal Response Function (mTRF) Toolbox: A MATLAB Toolbox for Relating Neural Signals to Continuous Stimuli. *Frontiers in Human Neuroscience*, 10.
- De Jong, N. H., Pacilly, J., and Heeren, W. (2021). PRAAT scripts to measure speed fluency and breakdown fluency in speech automatically. *Assessment in Education: Principles, Policy & Practice*, 28(4):456–476.
- Degutyte, Z. and Astell, A. (2021). The Role of Eye Gaze in Regulating Turn Taking in Conversations: A Systematized Review of Methods and Findings. *Frontiers in Psychology*, 12:616471.
- Ding, N. and Simon, J. Z. (2012). Neural coding of continuous speech in auditory cortex during monaural and dichotic listening. *Journal of Neurophysiology*, 107(1):78–89.
- Dreschler, W. A., Verschuure, H., Ludvigsen, C., and Westermann, S. (2001). ICRA noises: artificial noise signals with speech-like spectral and temporal properties for hearing instrument assessment. International Collegium for Rehabilitative Audiology. *Audiology: Official Organ of the International Society of Audiology*, 40(3):148–157.
- Erber, N. P. (1975). Auditory-Visual Perception of Speech. *Journal of Speech and Hearing Disorders*, 40(4):481–492.
- Gagl, B., Hawelka, S., and Hutzler, F. (2011). Systematic influence of gaze position on pupil size measurement: analysis and correction. *Behavior Research Methods*, 43(4):1171–1181.
- Giuliani, N. P., Brown, C. J., and Wu, Y.-H. (2021). Comparisons of the Sensitivity and Reliability of Multiple Measures of Listening Effort. *Ear & Hearing*, 42(2):465–474.

- Gravano, A. and Hirschberg, J. (2011). Turn-taking cues in task-oriented dialogue. *Computer Speech & Language*, 25(3):601–634.
- Hadley, L. V., Brimijoin, W. O., and Whitmer, W. M. (2019). Speech, movement, and gaze behaviours during dyadic conversation in noise. *Scientific Reports*, 9(1):10451.
- Hazan, V. and Baker, R. (2011). Acoustic-phonetic characteristics of speech produced with communicative intent to counter adverse listening conditions. *The Journal of the Acoustical Society of America*, 130(4):2139–2152.
- Heldner, M. and Edlund, J. (2010). Pauses, gaps and overlaps in conversations. *Journal of Phonetics*, 38(4):555–568.
- Hjalmarsson, A. (2011). The additive effect of turn-taking cues in human and synthetic voice. *Speech Communication*, 53(1):23–35.
- Hoeks, B. and Levelt, W. J. M. (1993). Pupillary dilation as a measure of attention: a quantitative system analysis. *Behavior Research Methods, Instruments, & Computers*, 25(1):16–26.
- Kahneman, D. and Beatty, J. (1966). Pupil Diameter and Load on Memory. *Science*, 154(3756):1583–1585.
- Kasthurirangan, S. and Glasser, A. (2005). Characteristics of pupil responses during far-to-near and near-to-far accommodation. *Ophthalmic and Physiological Optics*, 25(4):328–339.
- Koelewijn, T., Shinn-Cunningham, B. G., Zekveld, A. A., and Kramer, S. E. (2014). The pupil response is sensitive to divided attention during speech processing. *Hearing Research*, 312:114–120.
- Krueger, M., Schulte, M., Zokoll, M. A., Wagener, K. C., Meis, M., Brand, T., and Holube, I. (2017). Relation Between Listening Effort and Speech Intelligibility in Noise. *American Journal of Audiology*, 26(3S):378–392.
- Levinson, S. C. and Torreira, F. (2015). Timing in turn-taking and its implications for processing models of language. *Frontiers in Psychology*, 6.
- Li, H., Epps, J., and Chen, S. (2020). Think before you speak: An investigation of eye activity patterns during conversations using eyewear. *International Journal of Human-Computer Studies*, 143:102468.

- Mackersie, C. L., MacPhee, I. X., and Heldt, E. W. (2015). Effects of Hearing Loss on Heart Rate Variability and Skin Conductance Measured During Sentence Recognition in Noise. *Ear & Hearing*, 36(1):145–154.
- Mathôt, S. (2018). Pupillometry: Psychology, Physiology, and Function. *Journal of Cognition*, 1(1):16.
- McCloy, D. R., Lau, B. K., Larson, E., Pratt, K. A. I., and Lee, A. K. C. (2017). Pupillometry shows the effort of auditory attention switching. *The Journal of the Acoustical Society of America*, 141(4):2440–2451.
- McWalter, R. and McDermott, J. H. (2018). Adaptive and Selective Time Averaging of Auditory Scenes. *Current Biology*, 28(9):1405–1418.e10.
- Michael, T. and Möller, S. (2020). Effects of Delay and Packet-Loss on the Conversational Quality. In *Fortschritte der Akustik*, pages 945–948, Hannover, Germany.
- Miles, K., Weisser, A., Kallen, R. W., Varlet, M., Richardson, M. J., and Buchholz, J. M. (2023). Behavioral dynamics of conversation, (mis)communication and coordination in noisy environments. *Scientific Reports*, 13(1):20271.
- Petersen, E. B., MacDonald, E. N., and Josefine Munch Sørensen, A. (2022). The Effects of Hearing-Aid Amplification and Noise on Conversational Dynamics Between Normal-Hearing and Hearing-Impaired Talkers. *Trends in Hearing*, 26:233121652211033.
- Pichora-Fuller, M. K., Kramer, S. E., Eckert, M. A., Edwards, B., Hornsby, B. W., Humes, L. E., Lemke, U., Lunner, T., Matthen, M., Mackersie, C. L., Naylor, G., Phillips, N. A., Richter, M., Rudner, M., Sommers, M. S., Tremblay, K. L., and Wingfield, A. (2016). Hearing Impairment and Cognitive Energy: The Framework for Understanding Effortful Listening (FUEL). *Ear & Hearing*, 37(1):5S–27S.
- Plack, C. J., Barker, D., and Prendergast, G. (2014). Perceptual Consequences of “Hidden” Hearing Loss. *Trends in Hearing*, 18:233121651455062.
- Privitera, C. M., Renninger, L. W., Carney, T., Klein, S., and Aguilar, M. (2010). Pupil dilation during visual target detection. *Journal of Vision*, 10(10):3–3.
- Roberts, S. G., Torreira, F., and Levinson, S. C. (2015). The effects of processing and sequence organization on the timing of turn taking: a corpus study. *Frontiers in Psychology*, 6.

- Sacks, H., Schegloff, E. A., and Jefferson, G. (1974). A simplest systematics for the organization of turn-taking for conversation. *Language*, 50(4):696–735.
- Stone, M. A. and Moore, B. C. J. (2005). Tolerable Hearing-Aid Delays: IV. Effects on Subjective Disturbance During Speech Production by Hearing-Impaired Subjects:. *Ear and Hearing*, 26(2):225–235.
- Stone, M. A., Moore, B. C. J., Meisenbacher, K., and Derleth, R. P. (2008). Tolerable Hearing Aid Delays. V. Estimation of Limits for Open Canal Fittings. *Ear & Hearing*, 29(4):601–617.
- Summy, W. H. and Pollack, I. (1954). Visual Contribution to Speech Intelligibility in Noise. *The Journal of the Acoustical Society of America*, 26(2):212–215.
- Sørensen, A. J. M., Fereczkowski, M., and MacDonald, E. N. (2021). Effects of Noise and Second Language on Conversational Dynamics in Task Dialogue. *Trends in Hearing*, 25:233121652110244.
- Sørensen, A. J. M., Lunner, T., and MacDonald, E. N. (2024). Conversational dynamics in task dialogue between normal-hearing and hearing-impaired interlocutors. *Manuscript Submitted for Publication*.
- Theunissen, F. E., Sen, K., and Doupe, A. J. (2000). Spectral-Temporal Receptive Fields of Nonlinear Auditory Neurons Obtained Using Natural Sounds. *The Journal of Neuroscience*, 20(6):2315–2331.
- Van Rij, J., Hendriks, P., Van Rijn, H., Baayen, R. H., and Wood, S. N. (2019). Analyzing the Time Course of Pupillometric Data. *Trends in Hearing*, 23:233121651983248.
- Wagner, A. E., Nagels, L., Toffanin, P., Opie, J. M., and Başkent, D. (2019). Individual Variations in Effort: Assessing Pupillometry for the Hearing Impaired. *Trends in Hearing*, 23:233121651984559.
- Wendt, D., Koelewijn, T., Książek, P., Kramer, S. E., and Lunner, T. (2018). Toward a more comprehensive understanding of the impact of masker type and signal-to-noise ratio on the pupillary response while performing a speech-in-noise test. *Hearing Research*, 369:67–78.
- Wierda, S. M., Van Rijn, H., Taatgen, N. A., and Martens, S. (2012). Pupil dilation deconvolution reveals the dynamics of attention at high temporal resolution. *Proceedings of the National Academy of Sciences*, 109(22):8456–8460.

- Winn, M. B. and Teece, K. H. (2021). Listening Effort Is Not the Same as Speech Intelligibility Score. *Trends in Hearing*, 25:233121652110276.
- Winn, M. B., Wendt, D., Koelewijn, T., and Kuchinsky, S. E. (2018). Best Practices and Advice for Using Pupillometry to Measure Listening Effort: An Introduction for Those Who Want to Get Started. *Trends in Hearing*, 22:233121651880086.
- Yoo, K., Ahn, J., and Lee, S.-H. (2021). The confounding effects of eye blinking on pupillometry, and their remedy. *PLOS ONE*, 16(12):e0261463.
- Zekveld, A. A., Kramer, S. E., and Festen, J. M. (2010). Pupil Response as an Indication of Effortful Listening: The Influence of Sentence Intelligibility. *Ear & Hearing*, 31(4):480–490.
- Zekveld, A. A., Kramer, S. E., and Festen, J. M. (2011). Cognitive Load During Speech Perception in Noise: The Influence of Age, Hearing Loss, and Cognition on the Pupil Response. *Ear & Hearing*, 32(4):498–510.

Appendix

Appendix A

Supplementary Material

A.1 Python class for the implementation and removal of delay in live conversation

```
class DelayTracker:
    def __init__(self, win, audio_stream):
        self.win = win
        self.stream = audio_stream

        self.StateTracker = state.StateTracker(self.win, self.stream)

        "Set up delays"
        self.delay_input = 0
        self.delay_to_add = 0
        self.min_silence_dur = 0.2 * self.stream.rate / self.stream.chunk
        self.silence_dur = np.zeros((2, 1))
        self.speak_dur = 0
        self.delayed_silence_dur = 0
        self.zero_flag = 0
        self.delayed_pointer = 0
        self.delay_type = 'V'
        self.delay_low_val = 0
        self.delay_high_val = 0
        self.delayed_chunk = 0
        self.delay_buffer_size = 0.2 * self.stream.rate
        self.ongoing_delay_remove_flag = 0

        "Set up time history of input signals"
        self.input_history = np.zeros((self.stream.rate*self.StateTracker.history_dur,
        ↵ len(self.stream.in_ch))) # samples

        "Set up interface connections"
        self.init_window_values()
        self.update_delay()
```

```

        "Set up some structure to save information about changing delays"
self.delay_info = [["Input Delay",
                    "Current Delay",
                    "Current Pointer",
                    "Delayed Pointer"]]

def init_window_values(self):
    """Initialize parameters from window and connect signal slots to functions"""
    # Set variable delay as default
self.win.variable_delay_checkbox.setChecked(True)
self.delay_type = 'V'
self.win.fixed_delay_checkbox.setChecked(False)
self.win.line_edit_delay.setDisabled(True)

self.StateTracker.cal_level[0] = float(self.win.doubleSpinBoxCalCh1.value())
self.StateTracker.cal_level[1] = float(self.win.doubleSpinBoxCalCh2.value())
print(self.StateTracker.cal_level[0])

# Update the Delay and Memory Length if button clicked
self.win.delay_update_button.clicked.connect(self.update_delay)
self.win.silence_calibrate_button_delay_ch1.clicked.connect(lambda x:
↳ self.calibrate_silence(self.stream.in_ch[0]))
self.win.silence_calibrate_button_delay_ch2.clicked.connect(lambda x:
↳ self.calibrate_silence(self.stream.in_ch[1]))
self.win.doubleSpinBoxCalCh1.valueChanged.connect(lambda x:
↳ self.fine_tune_calibrate_silence(self.stream.in_ch[0]))
self.win.doubleSpinBoxCalCh2.valueChanged.connect(lambda x:
↳ self.fine_tune_calibrate_silence(self.stream.in_ch[1]))

# Change from fixed to variable if checkbox is changed
self.win.variable_delay_checkbox.clicked.connect(self.delay_to_variable)
self.win.fixed_delay_checkbox.clicked.connect(self.delay_to_fixed)

def delay_to_samples(self):
    """Converts delay value from ms to samples"""
self.delay_input = int(self.delay_input / 1000 * self.stream.rate)

def update_delay(self):
    """Update the amount of delay, if delay mode is fixed, otherwise update the range of delay values"""
if self.delay_type == 'F':
    self.delay_input = float(self.win.line_edit_delay.text())
else:
    self.delay_low_val = float(self.win.line_edit_delay_low.text())
    self.delay_high_val = float(self.win.line_edit_delay_high.text())
    print(self.delay_low_val, self.delay_high_val)
    self.delay_input = np.random.uniform(low=self.delay_low_val, high=self.delay_high_val)

self.delay_to_samples()

def delay_to_fixed(self):
    """Adjusts the delay mode from variable to fixed, if the UI checkbox is ticked"""
if self.delay_type == 'F': # Enable check/uncheck functionality
    self.delay_to_variable()
    return

# Update UI elements

```

```

self.win.variable_delay_checkbox.setChecked(False)
self.win.line_edit_delay_low.setDisabled(True)
self.win.line_edit_delay_high.setDisabled(True)

self.win.fixed_delay_checkbox.setChecked(True)
self.win.line_edit_delay.setEnabled(True)

self.delay_type = 'F'

self.update_delay()

def delay_to_variable(self):
    """Adjusts the delay mode from fixed to variable, if the UI checkbox is ticked"""
    if self.delay_type == 'V': # Enable check/uncheck functionality
        self.delay_to_fixed()
        return

    # Update UI elements
    self.win.variable_delay_checkbox.setChecked(True)
    self.win.line_edit_delay.setDisabled(True)

    self.win.fixed_delay_checkbox.setChecked(False)
    self.win.line_edit_delay_low.setEnabled(True)
    self.win.line_edit_delay_high.setEnabled(True)

    self.delay_type = 'V'

    self.update_delay()

def calibrate_silence(self, ch):
    """Estimates a background noise level in the room to be used for online voice activity detection"""
    cal_thresh_scaling = 3 # mean + x std deviation
    avg_duration = 200 # 200 chunks ~= .5 second
    if self.stream.current_pointer > avg_duration * self.stream.chunk:
        self.StateTracker.cal_level[self.stream.in_ch_map[ch]] = cal_thresh_scaling *
        ↪ np.sqrt(np.mean(np.square(self.input_history[self.stream.current_pointer - (50 *
        ↪ self.stream.chunk):self.stream.current_pointer,
        ↪ self.stream.in_ch_map[self.stream.in_ch[0]].astype('float32'))))
        self.win.doubleSpinBoxCalCh1.setValue(self.StateTracker.cal_level[0])
        self.win.doubleSpinBoxCalCh2.setValue(self.StateTracker.cal_level[1])
        print(self.StateTracker.cal_level)
    else:
        print(r'Calibration can only be run while a stream is active and has been running for a short
        ↪ duration. Press Run and then retry.')

def fine_tune_calibrate_silence(self, ch):
    """Enable spin-boxes to slightly adjust the silence calibration level, without needing to
    ↪ recalibrate"""
    if not self.StateTracker.cal_level[self.stream.in_ch_map[ch]] ==
    ↪ self.win.cal_spin_boxes[self.stream.in_ch_map[ch]].value():
        self.StateTracker.cal_level[self.stream.in_ch_map[ch]] =
        ↪ self.win.cal_spin_boxes[self.stream.in_ch_map[ch]].value()
    print(self.StateTracker.cal_level)

def time_shift_input_history(self, data):
    """Rotates the input history circular buffer, removing the oldest chunk and saving the newest"""
    self.input_history[0:-self.stream.chunk, :] = self.input_history[self.stream.chunk:, :]

```

```

self.input_history[-self.stream.chunk:, :] = data

def callback(self, data) -> np.ndarray:
    """Audio Processing Callback to modify the delay value and output advanced or delayed signals"""

    outdata = np.zeros(data.shape)

    # Time shift our input history
    self.time_shift_input_history(data)

    # Compute delay amount in chunks
    self.delayed_chunk = int((self.delayed_pointer - self.stream.current_pointer) / self.stream.chunk)

    # If delay is variable, then draw a new random delay value if appropriate
    if self.delay_type == 'V' and (self.delay_low_val + self.delay_high_val) != 0:
        # Check for speech in this block
        self.StateTracker.vad(data)
        # Check the state of the conversation in this block
        self.StateTracker.check_state()

        # if a floor transfer has occurred (the right way) then we draw a new delay value
        if self.delay_to_add == 0 and self.StateTracker.floor_change == 1:
            # Randomly sample new delay from specified range
            self.delay_input = np.random.uniform(low=self.delay_low_val, high=self.delay_high_val,
            ↪ size=1)
            self.delay_to_samples() # convert to samples
            self.delay_input = self.delay_input - (self.delay_input % self.stream.chunk) # delay should
            ↪ be evenly divisible by chunk
            self.StateTracker.floor_change = 0 # Reset floor transfer tracker so this doesn't happen
            ↪ repeatedly
            print('New delay value: ', self.delay_input)

    # Assign the difference between the new and previous delay as the delay to add
    if self.delay_input != self.stream.current_pointer - self.stream.chunk - self.delayed_pointer:
        self.delay_to_add = self.delay_input - (self.stream.current_pointer - self.stream.chunk -
        ↪ self.delayed_pointer)

    # If there is no speech in the last chunk adjust the silence duration to account for that, otherwise
    ↪ reset it
    # if self.stream.speech_sig[0, self.stream.current_pointer - self.stream.chunk] < 1:
    for _, ch in enumerate(self.stream.in_ch):
        if not self.StateTracker.vad_history[-1, self.stream.in_ch_map[ch]]:
            self.silence_dur[self.stream.in_ch_map[ch]] += 1
        else:
            self.silence_dur[self.stream.in_ch_map[ch]] = 0

        # Keep a running tally of the silence durations
        self.StateTracker.silence_history[0:-1, self.stream.in_ch_map[ch]] =
        ↪ self.StateTracker.silence_history[1:, self.stream.in_ch_map[ch]]
        self.StateTracker.silence_history[-1, self.stream.in_ch_map[ch]] =
        ↪ self.silence_dur[self.stream.in_ch_map[ch]]

    # Implement delay here, first we need to make sure the correct talker has the floor, the other
    ↪ talker has been
    # silent for an appropriate amount of time, and that we actually need to add delay
    if (self.delay_type == 'F' or self.StateTracker.floor == 1) and \
        self.delay_to_add > 0 and \

```

```

        self.StateTracker.silence_history[self.delayed_chunk,
        ↪ self.stream.in_ch_map[self.stream.in_ch[0]]] >= self.min_silence_dur and not \
        self.zero_flag:

    # Add to a new variable a track of how many zeros we need to output
    self.zero_flag += self.delay_to_add

    # Reset the silence duration since we added delay
    self.silence_dur[0] = 0
    # We remove delay as we can, waiting for some buffer to avoid artifacts, unless we are removing
    ↪ sample by sample
    elif (self.delay_type == 'F' or self.StateTracker.floor == 1) and \
        ((self.delay_to_add < 0 and self.StateTracker.silence_history[self.delayed_chunk,
        ↪ self.stream.in_ch_map[self.stream.in_ch[0]]] >= self.min_silence_dur)
        or self.ongoing_delay_remove_flag):

    # Set this flag to track that we need to remove delay
    self.ongoing_delay_remove_flag = 1

    # compute the amount of delay being removed as either the minimum or current silence duration
    delay_being_removed = int(max(self.StateTracker.silence_history[self.delayed_chunk,
    ↪ self.stream.in_ch_map[self.stream.in_ch[0]]], self.min_silence_dur))

    # Move up the delayed pointer by the delay being currently removed
    self.delayed_pointer += min(delay_being_removed, abs(self.delay_to_add))

    # Adjust our ongoing delay to add value (which can be negative) to reflect that weve changed the
    ↪ delay val
    self.delay_to_add = min(self.delay_to_add + delay_being_removed, 0)

    # Add an entry to our delay log
    self.delay_info.append([self.delay_input,
                            self.stream.current_pointer - self.delayed_pointer,
                            self.stream.current_pointer,
                            self.delayed_pointer])

    # If we removed all the delay necessary, then note that and print to console
    if self.delay_to_add == 0:
        self.ongoing_delay_remove_flag = 0
        print("Delay Remove Complete, Current Delay: ", self.delay_input)

    # If we're not sending zeros to the output, we need to actually send mic signals to the output
    if not self.zero_flag:
        # Update the delayed pointer
        self.delayed_pointer += self.stream.chunk

        # return the output signal at the appropriate location
        outdata[:, self.stream.out_ch_map[self.stream.out_ch[0]]] =
        ↪ self.input_history[self.delayed_pointer - self.stream.current_pointer - self.stream.chunk -
        ↪ 1:self.delayed_pointer - self.stream.current_pointer - 1,
        ↪ self.stream.in_ch_map[self.stream.in_ch[0]]]

    # Otherwise just output zeros and adjust the delay left to add accordingly
    else:
        # If we've added all the delay we need to, make a note of that and print
        if self.zero_flag - self.stream.chunk <= 0:
            self.delay_input -= self.zero_flag - self.stream.chunk

```

```

        self.zero_flag = 0
        self.delay_to_add = 0
        print("Delay Add Complete, Current Delay: ", self.stream.current_pointer -
        ↪ self.delayed_pointer)

    # Otherwise adjust our amount of zeros left to output
    else:
        self.zero_flag -= self.stream.chunk

    # Log to our delay tracker that we've updated the amount of delay
    self.delay_info.append([self.delay_input,
                            self.stream.current_pointer - self.delayed_pointer,
                            self.stream.current_pointer,
                            self.delayed_pointer])

    if len(self.stream.in_ch) > 1:
        outdata[:, self.stream.out_ch_map[self.stream.out_ch[1]]] = data[:,
        ↪ self.stream.in_ch_map[self.stream.in_ch[1]]

    return outdata

def start(self):
    """Called upon starting the program, updates delay values according to interface"""
    self.update_delay()
    print('Running Delay Module')

def stop(self):
    """Called upon stopping the program, saves delay file and resets parameters"""
    with open(self.stream.file_name + ".csv", "w", newline="") as data_file:
        writer = csv.writer(data_file)
        writer.writerows(self.delay_info)
    assert data_file.closed

    # Reset parameters so they are ready for next trial
    self.delayed_pointer = 0
    self.delay_input = 0
    self.delay_to_add = 0
    self.silence_dur = np.zeros((2, 1))
    self.speak_dur = 0
    self.delayed_silence_dur = 0
    self.zero_flag = 0

    self.delay_info = [["Input Delay",
                        "Current Delay",
                        "Current Pointer",
                        "Delayed Pointer"]]

    self.input_history = np.zeros((self.stream.rate*self.StateTracker.history_dur,
    ↪ len(self.stream.in_ch))) # samples
    self.StateTracker.vad_history =
    ↪ np.zeros(((self.stream.rate*self.StateTracker.history_dur)//self.stream.chunk,
    ↪ len(self.stream.in_ch)))
    self.StateTracker.silence_history =
    ↪ np.zeros(((self.stream.rate*self.StateTracker.history_dur)//self.stream.chunk,
    ↪ len(self.stream.in_ch)))

    print('Stopping Delay Module')

```

A.2 Python class for a sub-module that tracks the state of a conversation over time

```
class StateTracker:
    def __init__(self, win, audio_stream):
        self.win = win
        self.stream = audio_stream

        # Set up state flags
        self.floor = -1
        self.floor_change = 0
        self.prev_floor_start_ind = 0
        self.floor_start_ind = 0

        # Definition of minimum turn and pause lengths
        self.cede_dur = 180 # ms
        self.cede_dur = int(self.cede_dur * self.stream.rate / 1000 / 128) # to blocks
        self.take_dur = 90 # ms
        self.take_dur = int(self.take_dur * self.stream.rate / 1000 / 128) # to blocks

        # Set up vad history
        self.history_dur = 1 # seconds
        self.vad_history = np.zeros(((self.stream.rate*self.history_dur)//self.stream.chunk,
        ↪ len(self.stream.in_ch)))
        self.silence_history = np.zeros(((self.stream.rate*self.history_dur)//self.stream.chunk,
        ↪ len(self.stream.in_ch)))

        # Vad parameters
        self.cal_level = [20, 20] # default background noise level
        self.silence_leeway = 0.2
        self.speech_leeway = 0.8

    def calc_energy_thresh(self, sig, ch):
        """Performs energy thresholding on the input signal"""
        return int(np.sqrt(np.mean(np.square(sig))) > self.cal_level[ch])

    def vad(self, data):
        """Simple energy based chunk-by-chunk voice activity detection"""
        if self.stream.current_pointer > self.stream.chunk:
            for i, ch in enumerate(self.stream.in_ch):
                self.vad_history[0:-1, self.stream.in_ch_map[ch]] = self.vad_history[1:,
                ↪ self.stream.in_ch_map[ch]]
                self.vad_history[-1, self.stream.in_ch_map[ch]] = self.calc_energy_thresh(data[:,
                ↪ self.stream.in_ch_map[ch]], self.stream.in_ch_map[ch])

    def check_state(self):
        """Check the state of the conversation and denote who currently has the floor, and set flags for
        ↪ transfers"""

        if self.floor == -1 and \
            np.sum(self.vad_history[-self.cede_dur:, 0]) < self.silence_leeway * self.cede_dur and \
            np.sum(self.vad_history[-self.take_dur:, 1]) > self.speech_leeway * self.take_dur:
            self.floor = 1
            self.floor_change = 1
            self.prev_floor_start_ind = self.floor_start_ind
```

```

self.floor_start_ind = self.stream.current_pointer - self.take_dur
print('Floor Transfer 0->1')

elif self.floor == 1 and \
    np.sum(self.vad_history[-self.cede_dur:, 1]) < self.silence_leeway * self.cede_dur and \
    np.sum(self.vad_history[-self.take_dur:, 0]) > self.speech_leeway * self.take_dur:
self.floor = -1
self.floor_change = -1
self.prev_floor_start_ind = self.floor_start_ind
self.floor_start_ind = self.stream.current_pointer - self.take_dur
print('Floor Transfer 1->0')

```