Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

PsychPitchShifter: A software tool for psychophysiological pitch shift experiments

Semester Thesis

Linus Rüttimann

December 28, 2015

Supervisor: Prof. Dr. Richard Hahnloser

Department of Information Technology and Electrical Engineering, ETH Zürich

Abstract

The present thesis presents a software tool for the conduction of psychophysiological pitch shift experiments that drastically simplifies the typical experiment setup. In a pitch shift experiment the subject's auditory feedback is manipulated so that it hears it's own voice shifted in pitch and it is then observed if the subject adapts to the pitch shift. Additionally the software provides a feature that allows time-variant pitch shifts, which can be used as a basis for a new class of experiments. Furthermore a small experiment has been conducted to test the functionality of the software.

Contents

Chapter 1

Introduction

Humans and Animals use sensory information to regulate their interaction with the environment. On the one hand sensory information is often essential to determine and locate a specific motor target, on the other hand sensory information is also used as feedback to control the actuators (muscles) during motor operations. For example, if somebody intends to grab an object, he uses visual sensory information to determine the location of the object and the target position of his hand, additionally he relies on visual information during the movement operation to compensate for deviations between the actual and intended hand position.

In engineering such control tasks occur frequently and solutions are investigated in the field of control theory. Optimal solutions have been found for many problems, but little is known how the brain solves such motor control problems. One way to investigate the brain's solution is to manipulate the sensory feedback and observe the effect on motor performance. Numerous studies investigated the effect of visual feedback manipulation on the performance in reaching tasks [13, 14] and the effect of pitch shifted auditory feedback on voice production [11]. In both cases it has been observed that subjects adapt to feedback manipulations, i.e. they correct their motor commands so that the perturbation is partially compensated. Interestingly the relative compensation is usually less than 100% and it declines with increasing magnitude of the perturbation. Hahnloser et al. [5] propose a model to explain this non-linearity. They assume that adaption is not driven by the auditory feedback directly (which is noisy) but rather by an optimal estimation of the produced pitch (or motor output in general) given the auditory feedback and a predicted output pitch stemming from a forward model of the vocal organ. Their model takes into account that auditory feedback may not be self-caused but is produced by other sources. Given that the feedback is not self-caused, the estimation should be independent of it. Given that the feedback is self-caused the optimal estimation is a linear combination of the feedback and the predicted output weighted by the sensory and motor variance. Considering this, the optimal estimation of the produced pitch in general is a linear combination of this two cases weighted by the probabilities of the feedback being self-caused and not self-caused respectively. According to this model the magnitude of compensation should be modulated by the magnitude of the sensory noise variance. The purpose of this thesis was to program a framework that can be used in pitch shift experiments and particularly in an experiment designed to test this hypothesis.

The general procedure in pitch shift experiment is that subjects sustain a vowel while they hear a pitch shifted feedback through headphones and their voice is recorded with a microphone. The feedback may be pitch shifted during the full trial or only during a short time period in the middle of the trial. It is important that the subjects only hear the manipulated feedback, therefore the attenuation of the real feedback should be as high as possible. In the case where the shift is applied from the beginning of the trial, it is then measured if (and to what percentage) the subjects adapt to the pitch shift within a few trials. In the case where the shift is applied in the middle of the trial it is measured if the subjects show an immediate reaction on the pitch shift by changing the pitch of their voice in the opposite direction (opposing reaction) of the applied pitch shift. This reaction is called *pitch reflex* in the literature. Consult [7] for a comparison of these two experiment designs. In the pitch reflex experiments it has always been observed that a certain percentage of the reactions are in the direction of the pitch shift (following reactions); however, Behroozmand et al. [2] reasoned that the percentage and relevance of these following reactions have been underestimated in older studies.

In all the studies that are known to the author the Eventide Eclipse Harmonizer or similar devices were used to perform the pitch shifting. Often additional audio devices like mixers, amplifiers, etc. have been used. One aim of this thesis was to provide a much simpler experiment setup, which consists of only a software tool, a computer with sound card, headphones and a microphone. The software developed in this thesis is a MATLAB function and is named PsychPitchShifter. It implements the recording, the pitch shifting, the real time playback and some additional features. Currently there are three pitch shift algorithms implemented from which one can be chosen, all of them are open source. This is in contrast to the Eventide Eclipse Harmonizer and other commercial devices, where the algorithm is unknown to the user, because it is intellectual property of the manufacturer. Pitch shifting and in particular real-time pitch shifting is a non-trivial operation, there is no optimal solution and specific implementations always trade off different features. It can not be excluded that the algorithm implementation has an impact on the result of a certain experiment and it is therefore advantageous if the implementation is known.

Chapter 2

Implementation

PsychPitchShifter is a MATLAB MEX function. The MATLAB MEX API (application programming interface) provides an interface between MATLAB and functions written in C/C++. Although MATLAB itself has integrated audio interface capabilities it has been chosen to use C++ to keep the audio latency as low as possible. The audio latency is the period of delay between when the audio signals enters the sound card and when it emerges from it and it should be below 20-30 ms, in order that it is not perceived by the subjects [9]. The latency can be divided in two independent parts, the hardware induced latency and the algorithm induced latency.

The following subsections give an overview over the individual parts and features of PsychPitchShifter.

2.1 Audio Interface

In PsychPitchShifter RtAudio is used for the sound card interface. RtAudio is a set of C++ classes that provide a common API for real-time audio in- and output. Internally RtAudio uses the ASIO driver protocol, therefore PsychPitchShifter only works with sound cards that support ASIO. Most professional sound cards support ASIO, however most standard sound cards don't. If no professional sound card is available ASIO4ALL can be used, that's a software that adds ASIO support to sound cards that don't natively support it, however it is recommended to use a sound card with native support.

The sound card converts the audio input signal into a digital signal with a certain sampling rate and writes it periodically into a buffer, whenever the buffer is full RtAudio calls a user defined handler function. The data in the input buffer can then be processed and the output data can be written into an output buffer. After the next call of the handler function the data in the output buffer is converted back into an analog signal. Based on this procedure the minimum hardware latency is equal to $2 * \text{bufferSize}/\text{sampleRate}$ seconds. PsychPitchShifter uses a sample rate of 44.1 kHz and a buffer size of 64 samples, this buffer size is the minimum that is supported by most sound cards. The smaller the buffer size is, the more frequent the handler function is called and the more difficult it becomes for the CPU to keep up, because there is a considerable processing time overhead in a function call. As this overhead is even bigger in MATLAB functions, larger buffers or even additional buffer stages would have been necessary, thus PsychPitchShifter is programmed in C++. Additionally there is another part of the hardware latency, whose magnitude depends on the sound card driver and hardware implementation and which is independent of any user settings. This part is referred to as soundcardLatency below and it has been measured to be around 5 ms on a Motu 4pre USB sound card. Thus the complete hardware audio latency with this sound card, sample rate and buffer size is around 8 ms.

2.2 Pitch Shifting

Pitch is a psychoacoustic measurement, the American National Standards Institute defines it as follows:

> "Pitch [is] that attribute of auditory sensation in terms of which sounds may be ordered on a scale extending from low to high. Pitch depends primarily on the frequency content of the sound stimulus, but it also depends on the sound pressure and the waveform of the stimulus." [1]

This definition is relatively conservative, but it makes apparent that it is not fully understood yet how the brain maps an auditory input signal to pitch. It also shall be emphasized that not every sound has a pitch. Examples are unvoiced speech (whisper) or clicking and hissing noises, which are produced with the tongue and lips rather than with the vocal chords.

For the discussion of pitch shifting of voice signals a simpler definition is convenient. According to the source-filter model of speech production [6] a speech signal $s(t)$ can be decomposed in two parts, an excitation signal $e(t)$ and a time variant vocal tract filter $h(t, \tau)$. The speech signal can then be written as the convolution of these parts:

$$s(t) = \int e(t - \tau) h(t, \tau) \, \mathrm{d}\tau$$

For voiced speech the excitation signal $e(t)$ is a spike train convolved with a glottal wave pulse (Figure 2.1 and 2.2) and the pitch can be defined as the inverse of the inter-spike interval. The vocal tract filter is time variant due to

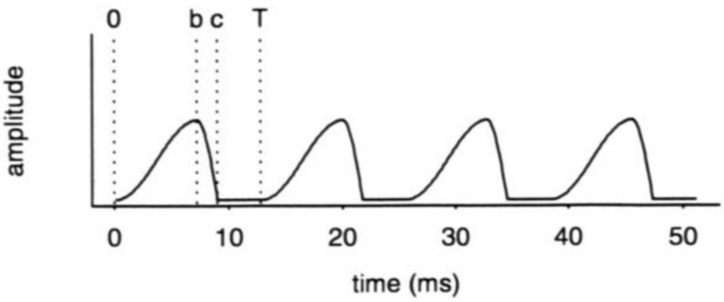

Figure 2.1: Example of a glottal pulse train $e(t)$ with constant inter-spike interval. Adapted from [6].

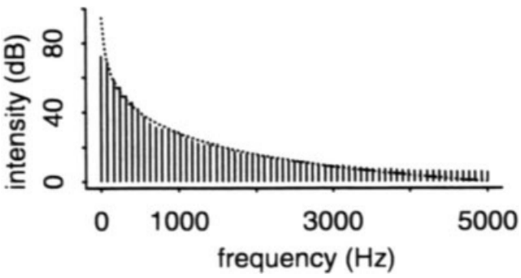

Figure 2.2: The spectrum of the pulse train shown in Figure 2.1. Adapted from [6].

the tongue and jaw movements, which modulate the geometry of the vocal tract and which thereby makes us able to vocalize different vowels. Furthermore it is stereotyped between people and does determine the timbre of the voice.

To shift the pitch one can not simply stretch the inter-spike intervals of $e(t)$, even though this does indeed shift the pitch it also changes the duration of the signal and thereby the rate of speaking. Instead the spike train must be split into sections of approximately constant inter-spike intervals (constant pitch) and then the spike trains in these sections must be replaced with spike trains of the same duration and stretched inter-spike intervals. The stretching factor is equal to the inverse shifting factor (the shifting factor is greater than one for upward shifts and smaller then one for downward shifts respectively). It is often convenient to state the magnitude of a pitch shift in a logarithmic unity, the measure cent is defined as:

$$i = 1200 \cdot \log_2(f) \text{ cent}$$

In which f is the shifting factor. 100 cent correspond to one semitone. As seen above, pitch shifting does not change the vocal tract filter, nevertheless many pitch shift algorithms do not only stretch the inter-spike intervals but also the vocal tract filter. By doing so the deconvolution of the speech signal into the excitation signal and the vocal tract filter can be omitted. The disadvantage of these algorithms is that they change the timbre of the voice. Algorithms that don't change the shape of the vocal tract filter are sometimes referred to as formant corrected pitch shift algorithms.

There are three different pitch shift algorithms integrated in PsychPitchShifter. All of them are based on a phase vocoder [4]. In a phase vocoder a short time Fourier transform is performed on an input signal to obtain a succession of overlapped spectral frames (analysis stage). The time delay at which every spectral frame is picked up from the signal is called the hop size. The spectral frames can than be manipulated. Afterwards a inverse Fourier transform is performed on the spectral frames and the frames are accumulated in a process called overlap-add (synthesis stage).

The first implemented algorithm is a traditional phase vocoder [4], it is named cpvPitchShifter. The second one is a modified phase vocoder [3], it is named smbPitchShifter. Both of them are rather simple pitch shifters, their C++ code is only around 150 lines long and they are not formant corrected. The main difference between these two algorithms is that in cpvPitchShifter the spectral frames are streched in the time domain, whereas in smbPitchShifter they are stretched directly in the frequency domain. Additionally smbPitchShifter tries to improve the frequency resolution by considering the phase changes between successive frames. The third algorithm is provided by the Rubber Band Library, which is an open source C++ library for time stretching and pitch shifting. It is more sophisticated then the first two algorithms and it also implements formant correction. Although the library is open source, the author of this thesis did not study the details of it's implementation, because the software is poorly documented and the source code is long. One disadvantage of the Rubber Band Library is that it produces hearable glitches at the onset of small pitch shifts. This is a considerable drawback for psychophysiological experiments, because it notifies the subjects about the onset of the pitch shift.

The latency measured in samples of phase vocoder based algorithms is equal to the frame size of the Fourier transform minus the hop size; however, the quality of the pitch shifting drops with a decreasing frame size, therefore it should not be chosen too small [8]. In PsychPitchShifter the hop size is set equal to the hardware buffer size, hence the accumulated hardware and algorithm induced latency in seconds is

$$\text{latency} = (\text{bufferSize} + \text{FFTframeSize})/\text{sampleRate} + \text{soundcardLatency}$$

As the frame size must be a power of two in the current implementation and the latency should be below 30 ms, the frame size should not be set greater than 1024 samples. With a fixed sample rate of 44.1 kHz, buffer size of 64 samples and the Motu 4pre sound card this corresponds to a latency of circa 30 ms. A smaller frame size can be chosen to decrease the latency, however this has a perceivable negative impact on the pitch shift quality.

2.3 Variable Pitch Shift

As mentioned in the introduction Hahnloser et al. [5] propose that the magnitude of the motor compensation in response to a pitch feedback manipulation is modulated by the magnitude of the sensory noise variance. In order to test this hypothesis experimentally there is a feature implemented in PsychPitchShifter that makes the pitch shift magnitude noisy. More precisely this features lets the pitch shift magnitude (measured in cents) follow a realization of a time discrete random process. This random process can be white noise; however, the resulting speech output sounds very unnatural and disordered. The pitch shift algorithms accept a new shift every 1.5 ms (determined by the hop size of the phase vocoder), however as depicted in section 2.2 the pitch of a voice signal can at most change once every glottal pulse, this corresponds to a period of 10 ms for a low pitch of 100 Hz for example, hence it makes no sense to change the shift magnitude more frequently. Additionally it has been observed that the pitch shift algorithms produce glitches at the transitions between two shift magnitudes, this glitches are most obvious for the Rubberband algorithm, but they are also perceivable for the other two algorithms. Because of this two reasons it makes sense to reduce the bandwidth of the random process and also limit the number of shift magnitude changes per time unit. To achieve this the following three options are available in PsychPitchShifter:

Configurable sampling interval:
: The sampling interval of the random process can be set to a integer multiple of the default sampling interval.

Noise filter:
: Filters the white noise with a second order butterworth low-pass filter.

Magnitude quantization:
: Quantizes the pitch shift magnitude with a configurable step size, to prevent very small shift changes that are not perceivable.

All of these options can be combined. The bandwidth of the random process is then given by:

$$B = \frac{f_c \cdot \text{sampleRate}}{2m \cdot \text{bufferSize}}$$

In which f_c is the cutoff frequency of the low-pass filter normalized to the nyquist frequency of the filter and m is the multiplication factor of the sampling interval.

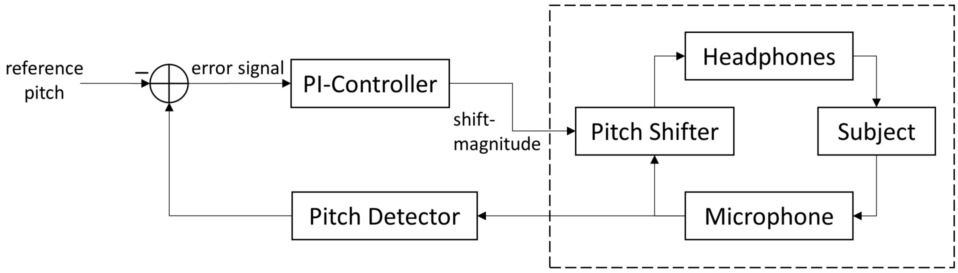

Figure 2.3: Block diagram of the control loop and experiment setup.

2.4 Pitch Control

If the assumption holds that the compensation to a pitch shift in a pitch reflex experiment is generally in the opposite direction of the applied shift and that the absolute compensation in cents is monotonically raising with the magnitude of the shift, one could eventually design an experiment, where the shift magnitude is dynamically modulated in order to drive the pitch of the subject's voice to a desired level by using an approach motivated by classical control theory. The objective of control theory is to control a system, so that its output follows a desired reference signal. To do this a controller is designed that monitors the output of the system and compares it with the reference signal. The deviation, which is called the error signal, is then applied as a feedback to the input of the system, to bring the actual output closer to the reference signal.

The stated assumptions are plausible after consideration of older studies like [11], where only a small fraction of the reactions were in the following direction and the magnitude of compensation was monotonically raising with the shift magnitude; however, this opposes the results of newer studies like [2]. Nonetheless PsychPitchShifter provides functionality to develop and possibly conduct such an experiment. In the accordant mode a real-time pitch detection algorithm is used to determine the produced pitch, then the error signal is calculated by comparing it with a constant reference pitch and a PI-controller is used to calculate the pitch shift magnitude. A PI-controller is a basic type of controller that calculates the system input as a linear combination of the error signal and its integral. The used pitch detection algorithm is described in [10], it was chosen because of its low latency and because a C implementation was already available. Figure 2.3 shows a block diagram of the control loop and experiment setup.

2.5 Additional Features & User Interface Description

PsychPitchShifter is designed as a non-blocking function, i.e. it returns immediately after the start of the recording. The recording can then be stopped with a second call of the function. During the recording the status can be polled, consult the help text (Appendix A) for a detailed description. PsychPitchShifter allows the experimenter to play a reference sound before the start of the trial. This can either be a synthesized e-piano sound of an arbitrary pitch or a custom sound. Furthermore the output signal can be mixed with pink noise, this may help to suppress the perception of the non-manipulated voice signal by the subject. The noise level can also be made adaptive to the input sound volume, so that the subject hears louder noise if it vocalizes louder.

PsychPitchShifter returns the recorded audio signal and also the pitch shifted audio signal. The recording starts immediately after the function call. A third variable returns the voice onset time relative to the recording start time. The voice onset is detected when the input sound volume exceeds a certain threshold that can be customized. For a detailed description for the usage of all the features and the user interface consult the help text (Appendix A).

Chapter 3

Test Experiment

3.1 Experiment Design

In order to test the software a small pitch shift experiment with 6 participants has been conducted. There were two conditions with 60 trials per conditions. In each trial the subjects heard a reference e-piano sound. The pitch of the reference sound did not change between trials and was chosen to be in a comfortable range for the subjects. After the offset of the reference sound the subjects were instructed to vocalize a vowel of the same pitch like the reference sound for 2.5 seconds. During the vocalization the pitch has been shifted for a duration of one second. The onset of the pitch shift was randomized between 500 and 900 ms after the voice onset. The shift magnitude was 100 cents (20 trials), -100 cents (20 trials) or 0 cents (20 trials). The sequence of the trials was randomized and there was a 0.7 s break between trials and a longer break between the conditions. In the first condition the shift magnitude was constant, in the second condition variable, meaning that the shift magnitude was a realization of a random process (as described in section 2.3). The standard deviation of the noise was set to 100 cents and the noise bandwidth to 7 Hz (the configurable sampling interval and magnitude quantization options were not used). In both conditions the subjects were instructed to not intentionally react to the pitch shifts.

3.1.1 Results

Figure 3.1 and 3.2 show the detected pitch averaged over the different trials and relative to the reference pitch (left hand side). The reference pitches were 150 Hz (subject 1, 2, 4 and 5), 120 Hz (subject 6) and 300 Hz (subject 3). Additionally the standard deviation is plotted (right hand side). For the pitch detection the algorithm presented in [12] was used.

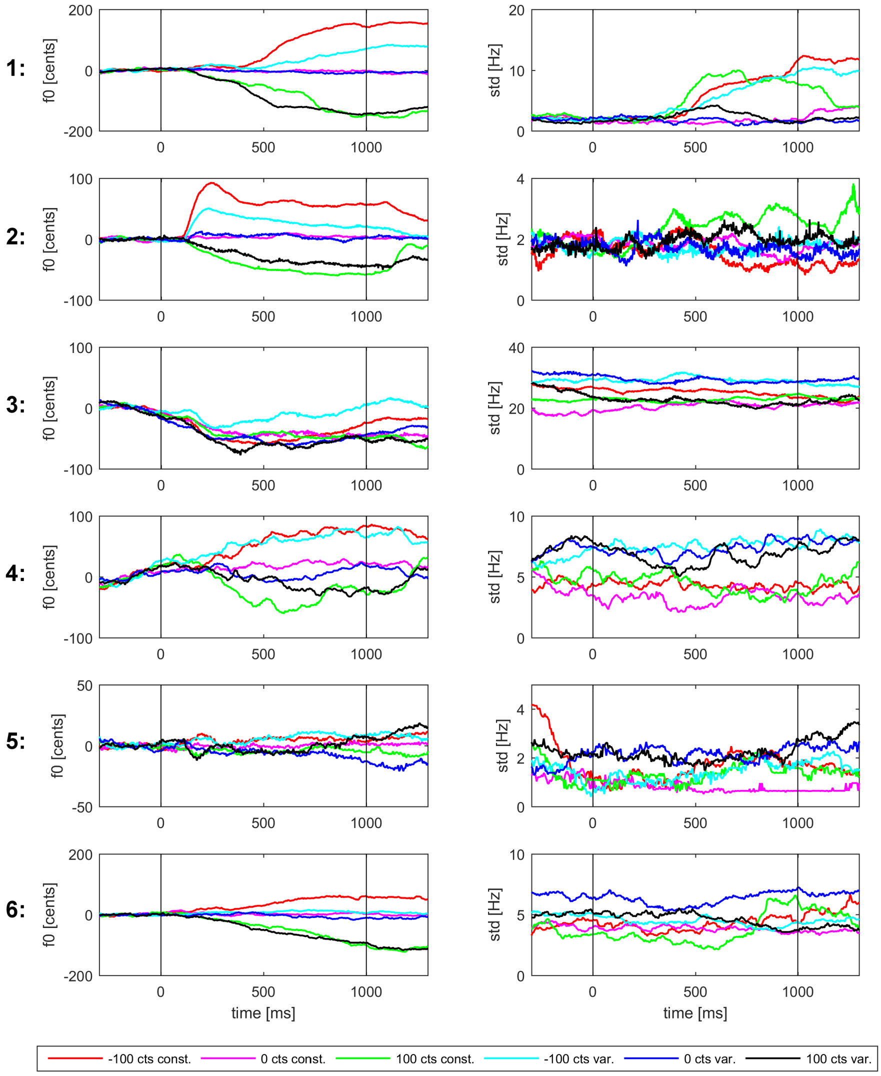

Figure 3.1: Plot of the pitch means (left hand side) and standard deviations (right hand side) for each subject. The vertical black lines indicate the on- and offset of the pitch shift.

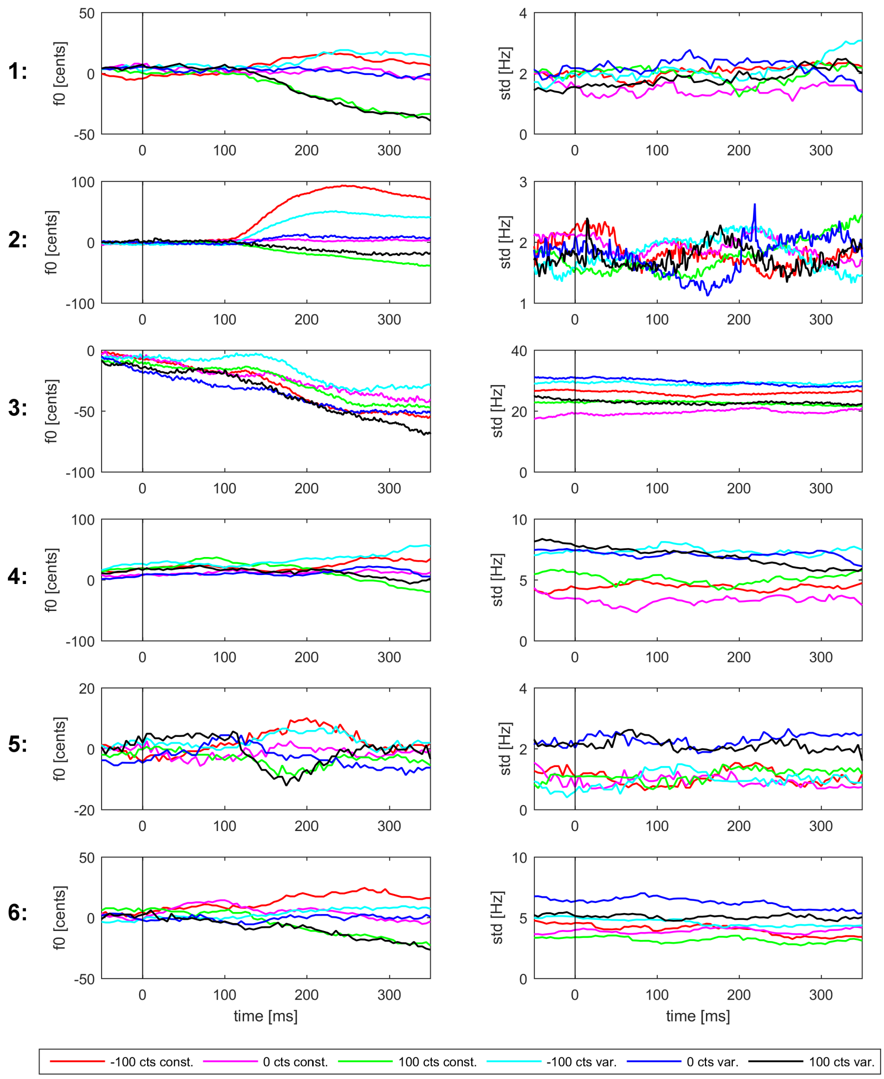

Figure 3.2: Zoomed version of Figure 3.1.

3.2 Discussion

The inter-subject variability is very high, furthermore subject three and four were not able to keep a constant pitch. While the response magnitude of subject two is very large and reaches almost 100% after 250 ms for the constant -100 cents stimulus, the response magnitude of subject five is smaller then 10% for all stimuli, which may not even be significant. The response magnitudes of subject one reach values above 100% after 600 ms, this may be due to a conscious response to the pitch shift, even though the subjects were instructed not to do so. Because of the small sample size and the big inter-subject variability it is however not possible to make generalized conclusions. The reaction latencies are between 100 and 200 ms, which is consistent with results from [11]. Considering only the data from subject two it looks like the response magnitude is smaller for the second condition (variant pitch shift), the data of the other subjects does not confirm this hypothesis though. For the data analysis the classical averaging technique has been used; however, for future experiments one should consider to use a better averaging technique presented by Behroozmand et al. [2]. They propose to presort the trials in following and opposing responses before the averaging, because they observed, as mentioned in the introduction, that a considerable percentage of the responses are in the following direction.

Chapter 4

Conclusion

The software developed in the context of this thesis is a convenient open-source tool for the realization of psychophysiological pitch shift experiments. It drastically simplifies the typical experiment setup by replacing dedicated audio hardware (mixers, effect devices, etc.) with an integrated software tool. Additonally it provides some features, namely the variable pitch shift and the pitch control feature, that can be used as a basis for a new class of experiments.

The conduction of a test experiment helped to improve the usability of the software. The existence of the pitch reflex could be confirmed; however, the sample size of 6 subjects is too small to make any further conclusions. A larger experiments must be designed to investigate if the pitch noisiness does indeed modulate the response magnitude.

Chapter 5

Acknowledgments

Thanks to the members of the songbird group, who helped me testing different experiment designs and who participated in the final test experiment. I especially want to thank Prof. Richard Hahnloser, who invested a lot of time to develop end test different experiment designs with me and who always had an open door for me to answer questions and discuss proposals.

Appendix A

PsychPitchShifter Help Text

```
PsychPitchShifter is a psychophysiological pitch shift experiment tool
   This function records audio and playbacks a pitch shifted version of it
   in real time. A fixed sample rate and internal audio buffer size is
   used:
       sampleRate: 44100
       bufferSize: 64
   All time parameters will therefore be rounded on integer multiples of
   bufferSize/sampleRate seconds.

   PsychPitchShifter(1,params):
       Starts a trial (recording and playback). params is a struct with
       the following fields:

       shifterId:      Determines the pitch shift algorithm.
                           0: smbPitchShift [default]
                           1: cpvPitchShift
                           2: rubberband
       windowSize:     The FFT window size of the Phase Vocoder. For the
                       rubberband algorithm the windowSize is not
                       adjustable and fixed to 1024.
                           0: 512
                           1: 1024 [default]
                           2: 2048
       deviceId:       Determines the audio device. Use the
                       PrintDeviceList auxiliary function to get a list of
                       possible IDs. Only ASIO devices are supported.
                       [default: 0]
       pitch_factor:   The pitch factor. [default: 1]
       play_ref_sound: Play a reference sound at the beginning of the
                       trial. [default: false]
```

```
ref_signal:     The reference signal. If not defined a synthesized
                e-piano sound will be played. [default: []]
ref_freq:       The frequency of the e-piano sound in Hz (only
                effective if ref_signal is empty). [default: 200]
ref_duration:   The duration of the e-piano sound in s (only
                effective if ref_signal is empty). [default: 1]
ref_amplitude:  The amplitude of the e-piano sound (only effective
                if ref_signal is empty). [default: 0.1]
voc_duration:   The duration of the trial in s, starting at the
                voice onset after the reference sound. [default: 1]
shift_onset:    The shift onset time in s relative to the voice
                onset time. [default: voc_duration/4]
shift_duration: The duration of the shift in s.
                [default: voc_duration/4]
shift_full_trial: The shift is applied during the full trial.
                [default: 0]
do_var:         Makes the pitch shift variable, it follows a
                realization of a low-pass filtered white noise
                stochastic process. [default: 0]
do_var_full_trial: Makes the pitch variable during the whole trial,
                not only during the shift. [default: 0]
std_dev:        the standard deviation of the white noise in cents.
                [defualt: 100]
fc:             the normalized cutoff frequency of the noise
                filter. Can have values between 0 and 1. Set it to
                1 to bybass the filter. [defualt: 0.01]
T_var:          Sets the sampling time of the stochastic process as
                a multiple of the frame rate
                (=sampleRate/frameSize). [default: 1]
var_quant_size: Quantizies the amplitude of the shifts to prevent
                very small changes in pitch. Set it to 0 to turn
                this off. [default: 0]
do_control:     Uses a PI controller that outputs an additional
                shift as an apttempt to drive the produced pitch
                to full compensation. [default: 0]
kp:             kp Parameter of the PI controller. [default: 0]
ki:             ki Parameter of the PI controller. [default: 1]
control_ref_freq: The reference frequency for th PI controller
                [default: ref_freq]
add_pink_noise: Adds Pink Noise to the output signal. [default: 0]
noise_gain:     The amplitude of the noise. [default: 0.005]
adaptive_noise_lvl: makes the noise aplitude adaptive to the
                input sound volume with the gain determined by
                noise_gain. [default: false]
```

```
    min_noise_lvl:  the minimum noise level in the adaptive noise mode.
                    [default: 0.005]
    max_noise_lvl:  the maximum noise level in the adaptive noise mode.
                    [default: 0.05]
    feedback_gain:  The amplitude of the output voice signal.
                    [default: 1]
    start_threshold: The threshold for the voice onset detection.
                    [default: 0.01]
    stop_threshold: The threshold for the voice offset detection.
                    [default: 0.01]

state = PsychPitchShifter(0):
    Returns a value according to the current state.
        0:  the function returns 0 till voc_duration seconds after the
            voice onset.
        1:  the function returns 1 after the end of the trial (which is
            voc_duration seconds after the voice onset), if the
            voice amplitude is still greater than stop_threshold.
        2:  the function returns 2 after the end of the trial, if the
            voice amplitude has reached a value smaller than
            stop_threshold.

[x,y,voice_on,stc_pf,var_pf,ctrl_pf,dpitch] = PsychPitchShifter(-1):
    Stops the audio recording and playback and returns the data.
        x:          The recorded audio data.
        y:          The pitch shifted audio data.
        voice_on:   The voice onset time in seconds.
        stc_pf:     The sequence (one sample per frame) of static
                    pitch factors.
        var_pf:     The sequence of pitch factor fractions produced by
                    the do_var feature.
        ctrl_pf:    The sequence of pitch factor fractions produced by
                    the do_control feature. Multiply all 3 factors to
                    get the accumulated applied pitch shift.
        dpitch:     The sequence of detected pitches. The do_control
                    feature relies on pitch detection, this is the
                    output of the internal pitch detection algorithm.
---------------------------------------------------------------------------
 (c) 2015 Linus Ruettimann (linus[dot]ruettimann[at]gmail[dot]com)
```

Bibliography

[1] ANSI. American National Standard Acoustical Terminology. *ANSI Sl*, pages 1–1994, 1994.

[2] Roozbeh Behroozmand, Oleg Korzyukov, Lindsey Sattler, and Charles R. Larson. Opposing and following vocal responses to pitch-shifted auditory feedback: Evidence for different mechanisms of voice pitch control. *The Journal of the Acoustical Society of America*, 132(4):2468, 2012.

[3] Stephan Bernsee. Pitch Shifting Using The Fourier Transform, 1999.

[4] Amalia De Götzen, Nicola Bernardini, and Daniel Arfib. Traditional implementations of a phase vocoder: the tricks of the trade. In *Proceedings Workshop on Digital Audio Effects (DAFx-00), Verona, Italy*, 2000.

[5] Richard HR Hahnloser, Gagan Narula, Alex Pouget, Dina Lipkind, and Ofer Tchernichovski. Optimal adaption in pitch-shift experiments. Unpublished manuscript, 2015.

[6] Jonathan Harrington and Steve Cassidy. The Acoustic Theory of Speech Production. In *Techniques in Speech Acoustics*, pages 29–56. Springer Science + Business Media, 1999.

[7] Colin S. Hawco and Jeffery A. Jones. Control of vocalization at utterance onset and mid-utterance: Different mechanisms for different goals. *Brain Research*, 1276:131–139, 2009.

[8] Nicolas Juillerat and Beat Hirsbrunner. Low latency audio pitch shifting in the frequency domain. In *2010 International Conference on Audio Language and Image Processing*. Institute of Electrical & Electronics Engineers (IEEE), 2010.

[9] Nelson Posse Lago and Fabio Kon. The Quest for Low Latency. In *Proceedings of the International Computer Music Conference*, pages 33–36, 2004.

[10] Eric Larson and Ross Maddox. Real-time time-domain pitch tracking using wavelets. *Proceedings of the University of Illinois at Urbana Champaign Research Experience for Undergraduates Program*, 2005.

[11] H Liu and CR Larson. Effects of perturbation magnitude and voice F0 level on the pitch-shift reflex. *J Acoust Soc Am*, 122:3671–7, 2007.

[12] Xuejing Sun. Pitch determination and voice quality analysis using Subharmonic-to-Harmonic Ratio. In *IEEE International Conference on Acoustics Speech and Signal Processing*. Institute of Electrical & Electronics Engineers (IEEE), may 2002.

[13] K Wei and K Körding. Relevance of error: what drives motor adaptation? *J Neurophysiol*, 101:655–64, 2009.

[14] C Wilke, M Synofzik, and A Lindner. Sensorimotor recalibration depends on attribution of sensory prediction errors to internal causes. *PLoS One*, 8:e54925, 2013.

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.