University of Dublin

Trinity College

*Pitch-Mod*
Pitch Modeling and Modification GUI

Mia de Burca
B.A. (Mod.) Computer Science, Linguistics and a Language
Final Year Project April, 2013
*Supervisor:* Prof. Ailbhe Ni Chasaide
Declaration

I hereby declare that this project is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university

______________________________ April 5, 2013

Mia de Búrca
Permission to Lend

I agree that the Library and other agents of the College may lend or copy this thesis upon request.

______________________________ April 5, 2013

Mia de Búrca
Acknowledgements

There are many people without whom this project would not have been possible.

Firstly, I would like to thank John Kane for his invaluable help and support over the past year, and for making an equally gleeful shout of joy when the code finally compiled and worked.

I am grateful to my friends, family, and partner for listening, often without understanding a word, and for their continual encouragement.

I would like to acknowledge my supervisor Prof Ailbhe Ní Chasaide’s insights and suggestions along the way.

A special thanks to my classmates who have made this challenging year enjoyable.
Abstract

This CSLL final year project was undertaken in conjunction with the Trinity College laboratory of phonetics and speech synthesis under the supervision of Professor Ailbhe Ní Chasaide. Under the optic of the creation of a tool for the research and instruction of speech pitch, the project involved the design and implementation of an interactive graphical user interface known as Pitch-Mod. This interface functions by modifying the pitch of a selected sample of recorded speech with values obtained from user input. Pitch-Mod is of great research value and its use in conjunction with speech training software is beneficial to second language learners, deaf persons acquiring spoken language and those suffering from specific language impairments such as autistic persons.
List of Figures

2.1 Summary of Intonation Functions .................. 7
2.2 Example of Tonal Language Pitch Contrasts ........... 7

4.1 MatLab’s GUIDE environment ......................... 15
4.2 The Pitch-Mod Template ............................ 16
4.3 Illustration of TD-PSOLA process .................... 19
4.4 Illustration of Glottal Closure Instants .............. 21
4.5 Illustration of SEDREAMS Process ................... 22
4.6 Pitch-Mod Launch Screen ........................... 24
4.7 Pitch-Mod state after LOAD ........................ 25
4.8 Pitch-Mod after DRAW ............................... 26
Contents

Declaration i
Permission to Lend ii
Acknowledgements iii
Abstract iv

List of Figures v

1 Introduction 1
  1.1 Aims ........................................... 1
  1.2 Overview ........................................ 1
  1.3 Motivation ...................................... 2

2 Background 3
  2.1 Introduction .................................... 3
  2.2 Prosody .......................................... 3
  2.3 Functions of Intonation .......................... 4

3 Review of Similar Software 8
  3.1 Introduction to Pitch Visualisation .............. 8
    3.1.1 Brief History of Functional Pitch Visualisation .... 8
  3.2 Pitch Visualisation Technologies ................. 9
  3.3 Advantages and Disadvantages of Visual Intonation Training 11

4 Project Development 14
  4.1 Introduction .................................... 14
  4.2 Design Choices .................................. 14
4.3 Development ........................................ 15
  4.3.1 Design ........................................ 15
  4.3.2 Pitch-Mod Code ................................. 16
  4.3.3 TD-PSOLA ..................................... 18
  4.3.4 Glottal Closure Instants with SEDREAMS .... 20
  4.3.5 Voicing Decisions ............................. 23
4.4 Interface Execution ................................. 24
4.5 Developmental Issues, Limitations and Improvements .... 26
  4.5.1 Contour Drawing .............................. 27
  4.5.2 Additional Features ............................ 28

5 Project Uses ........................................ 29
  5.1 Introduction ..................................... 29
  5.2 Research ......................................... 29
  5.3 Language Learning ............................... 30
    5.3.1 Foreign Language Learning .................... 31
    5.3.2 Other Language Learners ..................... 33
  5.4 Pitch-Mod in these Contexts .................... 36

6 Conclusion ......................................... 37
  6.1 Project Outcome .................................. 37
    6.1.1 Remaining Issues and Limitations ............ 38
    6.2 Future Work .................................... 38

Bibliography ........................................ 42
Chapter 1

Introduction

1.1 Aims

This project was undertaken with a view to developing an interactive graphical user interface (GUI) whose purpose is the alteration of pitch contours. The interactive nature of the GUI lies in the contours to be used in the process of modification as these are produced from user input. A primary goal of the development of this interface was the foundation of a tool that links the audio and the visual, capable of illustrating the relationship between subtle alterations in graphically represented speech contours and the corresponding perceived auditory changes. This functionality can be seen to have great practical uses in language acquisition by foreign language learners, deaf persons learning oral language and those seeking to overcome specific language impairments. Furthermore, the project has significant research implications in the domain of speech synthesis, analysis and auditory perception tests.

1.2 Overview

This dissertation, divided into six chapters, details the research and design of the Pitch-Mod interface. In Chapter 2, essential theoretical information will be provided to support the discussion of the main uses of Pitch-Mod and to provide a background understanding of the similar software and other techniques review in Chapter 3 when Pitch-Mod is situated within the field of pitch visualisation. Chapter 4 outlines the development of the Pitch-
Mod interface in MatLab and steps through the program execution by the user. The various uses for the Pitch-Mod interface are outlined in Chapter 5 and finally, in Chapter 6, the project outcome is discussed with reference to limitations, implementation challenges and possible future improvements.

1.3 Motivation

This project was motivated by a keen interest in foreign languages and research in the field of second language acquisition. As a student of foreign languages, the author is conscious of the multitude of difficulties the learner faces at all levels of competency. While much focus is directed towards vocabulary and grammar it has been the author’s experience that oral language skills are often neglected. Assisting learners in developing an awareness of pitch is one step towards promoting and supporting oral skills in learners.
Chapter 2

Background

2.1 Introduction

This chapter begins by defining prosody before moving on to a discussion of intonation, namely the functions of intonation in communication. This information provides a useful background for the consideration of the current project, the technologies review in Chapter 3 as well as the eventual uses of Pitch-Mod outlined in Chapter 5.

2.2 Prosody

Prosody refers to the supra-segmental characteristics of speech which arise from variations in stress, pitch and duration. They are features superimposed on the syllables that make up utterances (Ladefoged and Johnson, 2011) whose domain extends over more than one segment (Couper-Kuhlen, 1986). Voice modulations such as these provide essential foundations for intelligible and socially acceptable speech (Fourcin et al., 1993). The interest of this project lies in one specific aspect of prosody – the patterns of pitch that are known as intonation (Ladefoged and Johnson, 2011). While many consider intonation to involve not only pitch but other supra-segmental phenomena such as stress and pauses, Bolinger’s consideration of intonation as exclusively applied to gradient contrasts of pitch (Couper-Kuhlen, 1986) will be adopted in this project.
On an auditory level, intonation is related to the perception of pitch. What we hear as pitch translates roughly as the acoustic consequences of the fundamental frequency of speech or the vibration of air molecules set in motion by the vocal chords during voicing (Nolan, 2006). It is by no means a direct relationship as changes in pitch are not necessarily perceived at every change in fundamental frequency. Furthermore, true variation in pitch far exceed what is systematically relevant in a particular language (Couper-Kuhlen, 1986). Notably, since they are associated with the rate of vibration of the vocal fold, pitch changes can occur independently of stress changes (Ladefoged and Johnson, 2011).

2.3 Functions of Intonation

Spoken language, in contrast to written language, is capable of very subtle and meaningful variations which are relayed at times by the pitch of an utterance. English has a particularly high level of reliance on the use of pitch as a strategy for communicating the full meaning of what we say (Meyer, 1961).

"...demonstrates the economy of vocabulary, and of time required, for the conveyance of meaning in English, which results from our highly developed technique of making stress and pitch serve a semantic purpose" (Meyer, 1961, p. 11)

Intonation fulfills many communicational functions in spoken language. In tonal languages it is used to make phonemic distinctions (Meyer, 1961) while in other languages it conveys an array of information from paralinguistic to extra-linguistic and individual to discourse functions (Nolan, 2006). Elisabeth Couper-Kuhlen (1986), who considered that meaningful contrasts are what transmit information during communication, was one of many to attempt an exhaustive list of the functions of intonation. These functions are reproduced below with some supplemental examples.

Firstly, intonation has a grammatical function, determining sentence types and assisting in syntactic disambiguation by indicating the boundaries between grammatical units (Couper-Kuhlen, 1986). Frequently, the completion
of an intonation pattern coincides with the end of a grammatical clause however this is by no means a rule as this intonational marking may differ or even be absent, especially during fast speech (Nolan, 2006). The demarcation function of intonation aids listeners in building the correct sentence parse by indicating the nucleus position. This role is highlighted by ambiguous sentences, where more than one grammatical parse is available (Nolan, 2006).

1. *While eating my dog my cat and I watched television* (Nolan, 2006)

In contrasting different sentence types such as statements, commands, questions and responses, while there is no strict one-to-one mapping between modality and intonation pattern it is clear that rises and falls in intonation distinguish simple assertions from other sentence types (Nolan, 2006).

Tone-unit boundaries, the division of the message into chunks, and information structure, the grouping of these chunks in terms of new or given information is an informational function that can also be determined by intonation patterns. In the case of sentence 2 below, the postmodifying relative clause may be restrictive or non-restrictive, information determined by the kind of contour present (Couper-Kuhlen, 1986).

2. *That’s my brother who lives in Paris.*

New information or contrastive information is also highlighted by the position of the nucleus.

3. *I saw a MAN in the garden.* That a man was seen is new information (Couper-Kuhlen, 1986).

On yet another level, intonation signals illocutionary and attitudinal information about the speaker. Illocutionary force refers to the manner in which the speaker intends their utterance to be understood in a given context. Sentence 5 below could be interpreted as a plea, advice or a command (Couper-Kuhlen, 1986).

5. Shut the door (Couper-Kuhlen, 1986).

Attitudinal information aids the listener in determining how to interpret the speaker’s attitude towards the situation or their statement. It signals contrasts in irony and sincerity (Couper-Kuhlen, 1986), politeness and informality and psychological states such as anger, excitement or depression (Nolan, 2006). There is no denying the role of intonation in conveying attitude, consider the common observation cited by Nolan that the problem was “not what he said but the way that he said it” (Nolan, 2006, p. 19).

Intonation plays an important role in discourse regulating as it indicates the interactive aspects of linguistic organisation above sentence-level (Couper-Kuhlen, 1986). Turn taking in interactions is regulated by speakers making use of mechanisms to indicate that they have finished or wish to continue speaking (Nolan, 2006). More subtly, a speaker’s intonation pattern can indicate coherence or unrelatedness within a sequence of consecutive sentences (Couper-Kuhlen, 1986).

Furthermore, intonation provides indexical extra-linguistic information which involves contrasts which makes it possible to identify speakers as individuals or as members of certain social groups. Sex, age, socio-regional and occupational group membership can be determined from intonation - for example men typically avoid patterns and levels employed by women. Individual idiosyncrasies in speech patterns mean that individual speakers can also be identified at a distance (Couper-Kuhlen, 1986).
In many languages, including Thai (Thailand), Hausa (Nigeria), and Mixtec (Mexico), pitch is has a further use. Semantic differences between words can be made by phonemic contrasts, as is the case in English, or through the use of one of a predetermined set of unique pitch patterns which can be related to each syllable. Languages that employ this strategy are known as tone languages. While tone languages also use pitch for intonational purposes, they typically have less scope to elaborate an intonational system the greater their use of pitch in word contrast (Nolan, 2006).

As an example, in Mandarin Chinese a single syllable may be produced with four different pitches altering the semantic content of the word (Meyer, 1961, p. 9) as in the case below.

<table>
<thead>
<tr>
<th>Pitch Movement</th>
<th>Production</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>High- High</td>
<td>Rna</td>
<td>Mother</td>
</tr>
<tr>
<td>Mid - High</td>
<td>Rna</td>
<td>Numb</td>
</tr>
<tr>
<td>Mid - Low</td>
<td>Rna</td>
<td>Horse</td>
</tr>
<tr>
<td>High- Low</td>
<td>Rna</td>
<td>to Scold</td>
</tr>
</tbody>
</table>

This chapter has given an account of the theory underpinning the development of the Pitch-Mod program.
Chapter 3

Review of Similar Software

3.1 Introduction to Pitch Visualisation

A number of systems that implement functionality similar to that intended by the present work are discussed in this chapter, with reference to their benefits to the learner and possible faults. This will provide a technical context for the Pitch-Mod program.

During the course of this project other pitch manipulation and visualisation technologies were investigated. Among the programs described below, many are CALL (computer assisted language learning) interface systems with similar functions to the work undertaken for this project. While the majority of these tools were aimed at aiding learners in the acquisition of the pitch patterns of foreign language, some programs developed for other purposes were also encountered. A number of programs which take a somewhat original approach to the visualisation question were investigated also. This chapter begins with a brief introduction to the history of functional pitch visualisation followed by a review of some of the key projects which shed light on some of the pertinent considerations that go hand in hand with undertaking a project of this sort.

3.1.1 Brief History of Functional Pitch Visualisation

Since the 1960s there has been a rise in research surrounding the visualisation of pitch and the importance of visual clues in the process of language
acquisition (Germain and Martin, 2000; Levis and Pickering, 2004). Developments in the fields of linguistics and second language acquisition were crucial factors that led to more effort being invested into the use of computers to support the development of speaking and listening skills (Godwin-Jones, 2009). The common underlying idea of this approach is that learning occurs through the visualisation of a subject’s sound waveform or pitch contour, typically alongside that of a model utterance produced by a native speaker (Zinovjeva, 2005).

Initially, this research, undertaken by academics such as Anderson, Abberton and Fourcin (Spaai and Hermes, 1993), involved the use of pitch analysers and was geared towards the instruction of the deaf (Levis and Pickering, 2004). One of the originators of visualisation tools for second language learning was Vardanian, a TEFL teacher who used a cathode ray oscilloscope to display variations in melody over time of a model sentence and a speaker’s imitation (James, 1979; Germain and Martin, 2000).

### 3.2 Pitch Visualisation Technologies

Moving on to some more concrete examples of software and pitch visualisation systems, the SAID (Speech Auto-Instructional Device) system was developed by Lane and Buiten (1969). This was one of the original programs to enable the visualisation of the melodic curve, intensity and duration of a sentence pronounced by a user in real time (James, 1979). On a split screen, the deviations of the user’s production with regard to a template or model sentence were displayed. This kind of system was eventually determined to be too difficult and off-putting to the student to be successful, due to type of feedback concerns illustrated in section 3.3 (Germain and Martin, 2000).

The Intonation Meter, developed in the Netherlands, was a device originally designed for the hearing impaired. The particularity of the system was that it addressed the visualisation problem posed by interrupted voicing and micro-intonation perceptually irrelevant acoustic elements that impact on the displayed contour (O’Halpin, 2001). What it displays is a stylized pitch contour generated by performing interpolation on unvoiced segments leading to a simplification or smoothing out of the original pitch contour. This facilitates the process of interpreting graphical representation of pitch contours.
(Spaai and Hermes, 1993). It has been shown that this system is easier to interpret than models which do not process the extracted pitch contour as extra information, when left un-filtered, is distracting to learners. It can be used in conjunction with other intonation training programs (O’Halpin, 2001).

Another computer pronunciation training system was developed through a collaboration between UCD and Trinity; MySpeech. Their primary goal was the design and implementation of an easy, effective interface which guides the learner through the training process with extensive use of feedback to indicate errors. Pitch estimation and duration measures are used for evaluation then plots of acoustic, phonetic and articulatory features are displayed to assist users in the comprehension of their errors (Cabral et al., 2012).

Winpitch is a Canadian visualisation and synthesis tool designed through the collaboration of a phonetician/pedagogue and a phonetician/programmer. Germain and Martin (2000) were motivated by the objective of creating a system that would fast-track the development of auditory discrimination in learners and promote learning awareness. The result was a multi-functional program supporting the observation, visual and auditory comparison, segmentation, manipulation and redefinition of pitch contours. This system pioneered original functionality allowing the learner (or teacher) to record their productions, play them back at different speeds, visualise them and then annotate the contours with text. The student has the opportunity to become their own model by resynthesizing their own voice with some correct intonation contour (Germain and Martin, 2000).

Another example of a program that displays visual pitch curves is a product from Kay Elemetrics called Visi-Pitch. This is a commercial tool used by clinical speech therapists to help subjects suffering from language pathologies achieve therapy goals with visual feedback (Pentax, 2011). One of the fundamental components of Visi-Pitch is the “speech biofeedback”, a program for the extraction of acoustic parameter during speech production and their presentation in real time with intuitive visual displays. Patients are able to see both a target and their own pitch contour simultaneously on a split screen (Chun, 1998), similar to many of the programs described above.
An alternative to the CALL interactive GUI systems which targets deaf users is proposed by Boothroyd who discusses the design and implementation of a “wearable tactile sensory aid” (Boothroyd, 1985, p. 111) aimed at providing deaf persons with access to the intonation patterns of speech via touch. This system extracts the fundamental frequency of speech then represents it as the locus of pitch-synchronous vibratory stimulation of the skin. For the semi-hearing, the provision of a contour is a supplement to lip-reading that has already been shown to dramatically increase speech production performance; their hope is for this project to recreate this interpretation aid for the totally deaf (Boothroyd, 1985).

Yet another use of pitch visualisation interfaces lies within the area of research. PIT is a unique purpose application designed with the intention of integration into or cooperation with other applications. This software permits the visualisation and editing of up to 6 pitch contours within a single window. Operations can be performed on these contours such as editing, measurement, justification and splitting (Cioni, 1994). Based on this specific functionality set, it can be seen that this program was primarily implemented with a view to being a research aid for speech laboratories rather than a CALL application.

3.3 Advantages and Disadvantages of Visual Intonation Training

Through the observation of systems such as these which seek to make pitch explicitly accessible through visual representation it is clear that they give rise to a variety of problems, the most commonly reported of which relate to the unreliability of the measurement of pitch and the cost of operating the systems (Spaai and Hermes, 1993).

With regard to the systems which make use of a “feedback” component, there are some recurring problems. Often, there is no processing of the feedback type; the learner is simply presented with a measurement of their pitch. This is specialist information that is not at all illuminating to a typical language learner and guidance in the interpretation of this feedback is frequently insufficient (Zinovjeva, 2005). Moreover, the presence of voiceless
items and other perceptually irrelevant information is rarely accounted for leading to even more difficulty in understanding the visual feedback. Finally, pitch movements are not often given in relation to the corresponding sound onsets required for correct relation of audio and visual elements.

Spaai and Hermes (1993) were the first to attempt to address the problems caused by unvoiced sequences. Later, to overcome this issue of contour legibility, Germain and Martin (2000) included a “trace” function which plugs these gaps caused by unvoiced consonants.

"la fonction de traçage de WinPitch LTL permet de styliser le schéma mélodique de la phrase ou du discours pour mieux souligner, aux yeux de l’étudiant, les contours prosodiques pertinents “ (Germain and Martin, 2000, p. 68).

These researchers, who developed WinPitch, also introduced original functionality to take yet another problems into account. Their program attempts to assist users in the comprehension of pitch feedback by permitting contour annotation; teachers (or students) can mark vowel onsets, align contours to lexical units and highlight semantically important pitch changes - all information crucial to the interpretation of contours (Germain and Martin, 2000).

Some other researchers have experimented with game-like interfaces to resolve the issue of interpreting visual displays. In one program, speech input from the student is used to control movements in a game using a racing car. In order to stay on the road the user must closely imitate a model utterance; degrees of separation from the model correspond to movements off the centre of the road (Godwin-Jones, 2009).

A number of general advantages pertaining to the CALL systems should also be taken into account. These include the control, by the learner, of the pace of learning and number of production attempts, and the individualized attention learners receive from the private computer-learner relationship which is also less nerve-racking than corrections by a teacher in front of a class. Computer-based training has the advantage of supplying many more native speaker models than the teacher’s voice alone, an effect which has been shown to benefit pronunciation training. Furthermore, learner data can be accumulated to provide an invaluable source of information to researchers.
seeking to make improvements to language software and optimise approaches to pronunciation training (Godwin-Jones, 2009).

Germain and Martin (2000) also raise important points with regard to the advantages of intonation visualisation to learners. According to them, there is a body of research demonstrating that interaction is an important factor in the optimisation of a learner’s performance. Martin conducted further research indicating benefits from this approach. In a study using three control groups where the first did only traditional repetition intonation training, the second was shown a graphical representation of the model pitch contour during repetition and the third had the above and visual feedback of the speakers’ productions, results indicated that the visual reinforcement gained from comparison with the model speaker was a method of learning vastly superior to both others where improvements were judged by native speakers (James, 1979). De Bot (1982) demonstrated that audio-visual feedback is more effective in intonation training than auditory feedback alone. Along with Mailfert he showed that a 45-minute training session in the perception of intonation resulted in a statistically significant improvement in the production of English intonation by Dutch and French participants (Chun, 1998).

As many of these systems endeavour to provide students with an accurate visual representation of supra-segmentals in real time paired with auditory feedback, it is noteworthy that there are a number of factors influencing the benefit of audio and visual training in improving a learner’s perception and production of intonation. While the majority of recent studies demonstrate that this approach is effective, the relationship between perception and production is as yet not fully understood (Chun, 1998).

This chapter has outlined the significance of visualisation to pitch training for perception and eventually production, making reference to a number of systems and other technologies. In the following chapter the discussion turns to the particularities of the current project; interactive pitch contour modelling, modification and synthesis.
Chapter 4

Project Development

4.1 Introduction

This chapter outlines the development of the software beginning with a description of the code and then demonstrating its execution by a user.

4.2 Design Choices

A visual aid for speech analysis and training is a system that transforms selected acoustic information about speech into the visual domain in a way that is useful and ergonomically considerate. Creating a system that is viable, reliable and comprehensive is a substantial challenge (Povel and Arends, 1993). This project was undertaken with the restricted aim of enabling the manipulation of pitch contours selected by user input with a complementary visualisation aspect whose benefits will be seen in chapter 5.

This project was developed using the programming language MatLab, a high-level language and interactive environment for numerical computation, visualisation and programming. The major motivating factor of this choice of development language was the possibility for integration into the current TCD system implemented mainly in MatLab. As a development language, MatLab is a research standard in the field of signal processing due to its in-built functionality - the signal processing toolbox provides algorithms for analog and digital signal processing and can be used to visualise and play signals as well as perform computations such as Fast Fourier Transform among
others. Many speech processing algorithms are at least initially designed in MatLab. Furthermore, interface development in MatLab is facilitated by the GUIDE environment; graphical user interfaces can be created interactively and tuned programmatically as was the case for Pitch-Mod.

4.3 Development

In this section, the design of Pitch-Mod is outlined before the code itself is discussed in detail. TD-PSOLA, an algorithm developed by Moulines and Charpentier and implemented for the TCD laboratory by John Kane, and the GLOAT toolbox, a code package developed by Thomas Drugman, both of which make up the heart of the pitch modification process, are then described.

4.3.1 Design

The first step in the development of the Pitch-Mod programs was the design of an interface shell with MatLab’s GUIDE environment. This environment enables users to simply and quickly build the surface, graphical element of an interface using drag and drop functions. As can be seen in figure 4.1 below, options such as pushbuttons, text boxes and plots appear on the left and can be arranged along the axes of the new interface window.

![Figure 4.1: MatLab’s GUIDE environment](image-url)
Based on this graphical representation GUIDE automatically generates initialisation code and the skeletal code of empty callback functions which must then be implemented by the user to obtain the desired functionality. The design template developed for Pitch-Mod is illustrated below in figure 4.2. The implemented code for the current project is given in Appendix A, key elements of its callback functions are explained in section 4.3.2 below with references to lines of code as they appear in the appendix.

![Figure 4.2: The Pitch-Mod Template](image)

### 4.3.2 Pitch-Mod Code

After the initialisation code, a generic set-up of the initial launch-time parameters of the interface from lines 1-30 the input and output plots are defined at line 37-50. The first GUI function which is LOAD which requests from the user the location of a .wav file to input into the program. The user’s selection is then manipulated to extract Glottal Closure Instants (Line 83) and Voicing Decisions (78), parameters discussed in sections 4.3.4 and 4.3.5, as well as sampling frequency and fundamental frequency (Line 79). These values are then stored for global access as they are required for vi-
sualisation and essentially, for the eventual manipulation of the sound file. A spectrogram then is plotted to the interface’s lower output window (Line 95), using John Kane’s plot_spectrogram method (Appendix B), an implementation building on MatLab’s existing specgram function. This displays the spectrum of frequencies present in the speech signal as they vary over time with amplitude indicated by varying degrees of darkness. A broadband spectrogram was preferred as its short duration captures rapid changes in amplitude allowing for the observation of excitation of the vocal folds during voiced speech however time resolution was kept at a level limiting spectral smearing so that the movements of harmonics can still be seen.

Next the user defines the pitch contour to be used via the DRAW pushbutton. This occurs from lines 113-118 by means of a while loop conditioned by mouse-point clicks where left clicks select Cartesian points on the input plot and the user signals an end to the contour “drawing” with a right click. The user’s input is then normalized and altered for passing to the TD-PSOLA function. First, the input is forced to follow x-axis linearity (the representation of positive progression of time) by sorting the input array (Line 120-121). Then, provided certain conditions are met, these points are interpolated for finer plotting detail and to force correspondence to the length of the sound file (Line 135). They are then plotted onto the input window as a curve (Line 136). Next, the y-axis points, which correspond to pitch values, are extracted from the input and aligned with GCI elements (Line 143). This information is stored to the handler in advance of the modification function.

The information obtained from the LOAD and DRAW functions is accessed when the user selects the MODIFY pushbutton. In this function, the vector of pitch values generated by the user and other parameters calculated from the input file are passed to the TD-PSOLA function (Line 166) discussed in detail below. Next, the PLAY call-back first determines whether any pitch alteration has occurred previously based on the content of the nummod global variable (Line 181). Before any modification has occurred, the original sound file can be played. If modification has taken place, PLAY accesses the values of the modified sound file and plays back to the user. Finally, RESET causes the GUI to return to its initial state (Line 216); no stored information relating to modifications is retained and the user is free to attempt more modifications with the same or different sound files.
Later, a STORE information option was included in the GUI. It was considered that users might require access to pitch values created with Pitch-Mod after exiting the interface. While, MatLab permits the sending GUI variables to the command line however storage in a .txt file was preferred as it has the benefit of being accessible by other programs also. Pitch-Mod enumerates the modifications during execution (Line 169) and uses this information to label output pitch values. On execution of the STORE push-button, the values produced by the user to represent the pitch are first multiplied by the average fundamental frequency (Line 198) before storage in the output file (Line 205). This additional functionality permits the user to save work, a benefit to both researchers and learners.

4.3.3 TD-PSOLA

Pitch-Synchronous Overlap and Add (PSOLA) is a family of widely used second generation signal processing techniques (Taylor, 2009) which can be used to manipulate the pitch of a speech signal (Lemett, 1999). The foundation of all PSOLA algorithms involves the isolation of pitch periods of a signal and their re-synthesis via an overlap-add operation after certain modifications. More specifically, the input signal is decomposed into separate but overlapping segments, a series of elementary waveforms (Schnell et al., 2000) which are then recombined to make desired pitch and/or duration alterations. The attraction of PSOLA is that these modifications can be performed without any adverse effect on other speech sound attributes such as voice quality or formant frequencies (Taylor, 2009).

TD-PSOLA, the most popular of these algorithms for pitch and timing adjustment, operates by means of the following steps. First, an epoch detector is used to find instants of glottal closure, discussed in section 4.3.4. Next, the input signal is segmented by windowing around each individual pitch period with a Hanning window. Typically, segments extend to one pitch period before and after the epoch which is necessary to allow for lowering of the pitch. A set of synthesis epochs, which are differently spaced than the originals, are generated from the f0 and timing specifications of the original signal. It is the mapping of the original epoch information onto the synthesis epochs that causes pitch modification – if epochs are closer together the effect is a higher pitch and conversely, further apart epochs generate a waveform
of lower pitch. The new frames are recombined using an overlap and add function to create the final signal (Taylor, 2009).

Figure 4.3: Illustration of TD-PSOLA process (Taylor, 2009, p. 429)

With regard to the epoch to epoch mapping function, there are several possible options from linear functions to more sophisticated functions which have the benefit of preserving transition dynamics. It should also be noted that modifying pitch has the bi-product of changing duration, since the dis-
tance between epochs is being changed. This effect must be compensated for using frame duplication or elimination. (Taylor, 2009).

While the speech waveform produced by TD-PSOLA is not precisely the same as the original, provided the modification settings are within certain moderate bounds it is close to being perceptually indistinguishable since no local modification of the frames has occurred, only a rearrangement of the positions (Taylor, 2009). For very large modifications, artefacts can be introduced into the signal. This algorithm acts directly on the signal without relying on any model, with the result that there is no loss of detail in the signal. Furthermore, there is no explicit source/filter separation meaning that independent control of pitch duration and signal formants is possible (Schnell et al., 2000; Taylor, 2009).

As TD-PSOLA works pitch-synchronously, it is essential that the pitch periods be located with great accuracy. In this project, the pitch periods were centred on instants of glottal closure as detected by the SEDREAMS function, detailed in section 4.3.4. In reality, segments may be based on any instants provided that they lie at the same relative position in each frame (Lemetti, 1999).

In terms of performance, the synthesis quality of TD-PSOLA is considered extremely high (Schnell et al., 2000; Lemetti, 1999), provided that the speech has been accurately segmented and modifications are not excessive. TD-PSOLA is judged similarly favourably with regard to execution time (Lemetti, 1999) and it has a low computational cost since signal transformation is produced by time manipulation alone (Schnell et al., 2000); the main computational load is incurred during glottal closure instant detection.

4.3.4 Glottal Closure Instants with SEDREAMS

Glottal closure instants (GCI), often referred to as epochs, relate to the instant of significant excitation of the vocal tract. These time points are considered to correspond to the moments of high energy in the glottal signal during voiced speech (Drugman and Dutoit, 2009). GCIs have considerable perceptual importance, given their influence on the pitch of oral production,
and knowledge of their whereabouts is of particular value to speech processing (Drugman et al., 2012).

Figure 4.4: Illustration of Glottal Closure Instants

The TD-PSOLA method functions by exploiting the pseudo-periodicity of speech which is not possible without knowledge of the precise location of glottal closure instants. In their paper reviewing state-of-the-art GCI decision algorithms Drugman et al. (2012) detail the relative functionality and efficiency of five of the most effective of these and determine that, along with YAGA (Yet Another GCI Algorithm), SEDREAMS (Speech Event Detection using Residual Excitation and a Mean-Based Signal) performs best in terms of clean speech and robustness faced with additive noise.

SEDREAMS is an automatic method which uses smoothing and measures of energy in the speech signal for GCI detection. It is considered an accurate and robust algorithm for locating GCIs (Drugman et al., 2012). SEDREAMS is a two part method in which a mean-based signal is first determined to limit the signal to discrete intervals of presence before refinement is done to locate the precise positions of the GCI within each interval (Drugman et al., 2012).
The mean-based signal is calculated by obtaining the mean of sliding windowed speech segments (Drugman and Dutoit, 2009). While the window type is of little importance, its length has a considerable impact on the false alarms and misses that may occur. Next, within each of the defined intervals, the largest LP residual is pinpointed as it is assumed that the largest discontinuity of the signal corresponds to the GCI (Drugman et al., 2012).

Since oscillation of the signal occurs at the local pitch period, the signal guarantees good performance in terms of reliability ensuring limited risks of misses or false alarms. This is one advantage of using a mean-based signal. Furthermore, the GCI timing error is bounded by the length of the intervals of presence that are derived from this signal. Smoothing is advantageous because the vocal tract resonances, additive noise and reverberation are attenuated while the periodicity of the speech signal is preserved (Drugman et al., 2012).

Below is an illustration of the GCI detection process employed by SEDREAMS where (a) is the speech signal, (b) is the mean-based signal, (c) the intervals of presence derived from it, (d) is the LP residual signal and (e) is the synchronized dEGG with the GCI positions located.

![Figure 4.5: Illustration of SEDREAMS Process](image)

Figure 4.5: Illustration of SEDREAMS Process
4.3.5 Voicing Decisions

Another aspect central to the TD-PSOLA method relating to the resynthesis process is the voicing decision. The boundaries between voiced and unvoiced speech are important to synthesis in terms of modelling the voice source; unvoiced speech does not carry intonational information. As TD-PSOLA relies on knowledge of the GCI and unvoiced speech does not cause any points of excitation it is important to have knowledge of its whereabouts so as not to introduce GCIs where there are none. Though many approaches exist, for the purpose of this project, voicing decisions will be determined using harmonic information in the residual signal, a method designed for pitch tracking for which voicing decisions is a useful side effect.

First, an autoregressive modelling of the spectral envelope is estimated from the speech signal and the residual signal is obtained by inverse filtering. This has the advantage of removing a large part of the contributions of noise and vocal tract resonances. Then, for each windowed frame, covering several cycles of the residual signal, the amplitude spectrum is computed. The observed amplitude spectrum has a somewhat flat envelope however within voiced segments of speech it presents peaks at the harmonics of the fundamental frequency. When summation of residual harmonics (SRH) is performed, no particularly high values are observed during unvoiced regions of speech. It is in this way that SRH can be used to determine voicing decisions by a simple local thresholding; where the SRH is greater than a given threshold the segment is deemed voiced. (Note that the residual spectrum must be normalized in energy for each frame to allow for comparison with a threshold). This method described in Drugman and Alwan was demonstrated to have high performance results, in terms of voicing decision error (Drugman and Alwan, 2011).
4.4 Interface Execution

This section outlines the steps involved in the execution of the Pitch-Mod by a user.

On launching the program, the user is displayed the screen illustrated in figure 4.6 below. They must then select and load an audio file containing a speech sample.

![Pitch Mod Launch Screen](image)

Figure 4.6: Pitch-Mod Launch Screen

Based on this input, a spectrogram is generated and displayed in the lower output window as seen in figure 4.7.
Then, the user selects Cartesian points representative of pitch values with the mouse, these are then plotted as a curve on the output screen, see figure 4.8. At this stage, the user may select the button to apply their chosen contour on the speech sample. Now the user has the option to play back the modified sound file, store their generated pitch values, reset or exist the program.
4.5 Developmental Issues, Limitations and Improvements

There are many conceptual and technical problems associated with building visual aids for speech training. Concrete issues include how to extract speech parameters for analysis or manipulation and how to map these parameters onto the available visual display in a clear manner. All interface development is faced with ergonomically important questions with regard to user understanding and preference; arrangement of push buttons and figures for plotting, relationship between loading files and displaying information - nothing can be taken for granted. Finally, in technical terms, the implementation of this desired functionality into a properly working device is quite challenging (Povel and Arends, 1993).

During the course of this project a number of improvements and alterations were made to the original design concept, for a variety of reasons. In some cases it was determined through trial and error that the initial idea
would not be possible at time. In other cases, more appropriate methods were found to achieve the desired function. Certain limitations arose from insufficient knowledge of MatLab functionality, this programming language, used for the development of Pitch-Mod was unknown to the author before undertaking the project. For example, the specificities of MatLab’s complex matrix manipulation functions had to be learnt. Where initial implementation was at times cumbersome and inefficient, refinements were made along the way resulting in the code illustrated in Appendix A.

A number of problems were also incurred attempting to relate the GUIDE environment’s parameters to hardcoded values. GUIDE includes a menu, Property Inspector, designed to alter interface, pushbutton and plot parameters. However, when it came to labelling and setting the range of the input and output plot axes it was eventually determined that neither hardcoding nor alterations by Property Inspector alone were sufficient and a seemingly random combination of both was eventually settled upon. Unfortunately, despite many attempts it was not possible to maintain the desired ranges throughout execution - after calls to plot_spectrogram the x-axis values of the output graph jump to fit the image. If the input plot ranges were changed to correspond to these, errors were introduced into the modification process whereas if the output plot was restricted from making this jump the spectrogram was not displayed correctly. No solution was found for this issue.

4.5.1 Contour Drawing

A number of options were considered with regard to the manner in which the user would input the pitch values. The original proposal involved tracking the mouse point movement by using MatLab’s uiDraw function however it was found that this function was not suitable for the desired graphical illustration. MatLab automatically closes the space underneath a uiDraw contour, linking the first and last points. Furthermore, extracting values from objects created by uiDraw involved added complexity. Hence, the Cartesian coordinate system for data representation was settled on, a system in which the user places points with mouse clicks and the (X,Y) coordinates of which are easily accessed by the program. This method allows for clean efficient access of data for storage and manipulation later in the program and is user friendly. All that remained was to include a contour overlying and connecting these points but this was easily accomplished with the plot function.
A related question involved how to restrict users in their use of the DRAW method so as not to induce errors. For example, it was initially considered that users should be prohibited from entering points that move backwards on the x-axis (i.e. entering a point at x= 5 and then x = 3) as this would result in a curve moving backwards in time or looping over itself. However, it was later decided that this issue could be avoided by sorting the output vector of (X,Y) coordinates eliminating the time domain concern and allowing the user more freedom in their input choices. One restriction maintained imposes a minimum of two input points in order for pitch manipulation to occur.

4.5.2 Additional Features

During the development process, a number of additional features were imagined that, while not strictly necessary, complemented the interface functionality. The PLAY button was altered to allow users to play their selected audio track unaltered before they began modification; this allows for more ready comparisons. A store function was included which outputs the vector of pitch values, multiplied by the average f0 of the sample, along with a label indicating the alteration attempt to a .txt file. It was considered that users would benefit from access to their modification attempts outside of the Pitch-Mod program.

This chapter has described the development of Pitch-Mod from design to concrete implementation before providing an example of execution by the user. The next chapter relates to the tasks for which the user might employ Pitch-Mod.
Chapter 5

Project Uses

5.1 Introduction

Now that the particularities of the Pitch-Mod interface have been outlined its concrete uses will be discussed in this chapter. Pitch-Mod, as an interactive and intuitive system can be used for many purposes. In the domain of research it is an illustrative tool and with regard to language learning it is a user friendly training device both functions which will be described within this chapter.

5.2 Research

The original motivation for this project was to provide a simple effective interface for pitch modelling and modification to compliment the current system of the University of Dublin, Trinity College’s phonetics and speech laboratory with the possibility that the tool could eventually be integrated into their newly developed voice analysis system – GlóRí. Pitch-Mod has research implications which are of particular relevance to the research being conducted in the phonetics and speech laboratory relating to speech analysis and synthesis with a focus on voice qualities. Pitch-Mod can be used to perform perception tests or it may be used as a teaching aid for the many undergraduate or postgraduate classes conducted in the laboratory.

The integration of Pitch-Mod into the current TCD system is made possible due to its development in the MatLab environment. Similarly, the inter-
face, while it has been designed and implemented for fundamental frequency modification, can easily be updated to allow modification of different speech parameters such as formant frequencies, glottal source and voice quality settings. Its development in the Matlab environment facilitates this flexibility as, as noted elsewhere, a large proportion of speech processing research is done using Matlab. While this program’s functionality can be replicated easily by exploiting Matlab’s inline command window, the use of a GUI has the benefit of being user friendly, intuitive and simple. Furthermore, Pitch-Mod can be operated with no prior knowledge of Matlab, meaning that researchers can focus on the task at hand without getting side-tracked by the substantial task of acquiring new programming skills.

In terms of application, Pitch-Mod may be a useful means of performing perceptual tests. It can be used by researchers to assess the perceptual impact of modifying the pitch of a given utterance, since not all possible perceivable pitch events are of semantic relevance (Couper-Kuhlen, 1986) or to investigate the correlation between f0 and pitch. For example, a researcher may play speech samples to a participant while performing pitch modifications and make a note of these changes as well as the level of modification at which participants makes certain judgements. The interface could therefore be used to facilitate the assessment of the role of certain aspects of the speech signal in signalling functions such as focus, prominence, attitudes, and emotions.

Pitch-Mod is also a valuable teaching aid. Researchers at the TCD laboratory are frequently called upon to share their knowledge by lecturing undergraduate and postgraduate students of computational linguistics and clinical speech science among others. Pitch-Mod could provide a useful teaching aid in the illustration of different speech contours and their acoustic properties and in highlighting the distinction between changes in fundamental frequency and perceived pitch movement.

5.3 Language Learning

As a teaching tool Pitch-Mod is of great benefit to second language learners and those suffering from deafness or specific languages impairments. Both groups face specific challenges in their progression towards fluent oral speech
in a given language caused by inadequate understanding and incorrect production of intonation. Crucially, for reasons illustrated below, it is considered essential for the user to develop an awareness of pertinent pitch movements as it is hoped this will lead to better production skills. In this way communication challenges could be overcome.

5.3.1 Foreign Language Learning

Many consider intonation to be the first aspect of language that children acquire (Chun, 1998, Alazard, Astésano & Billières, 2010). From a very young age, infants respond to the prosody of their mother tongue, despite all other distinguishing cues being factored out (Alazard et al., 2010). While child L2 learners demonstrate little difficulty acquiring native-like prosody in the L2, this facility seems to disappear in adult learners who tend to find L2 prosody acquisition an impossible feat. Given this evident degree of extreme difficulty and the complexity of the task, prosody should long have been a main focus of second language teaching (Chun, 1998) however it was ignored for many years when methodologies favouring text based learning and translation reigned (Alazard et al., 2010). While it is still a frequently ignored skill (Spaai and Hermes, 1993), some instruction methods such as the structure-global-audio-visual method and the verbo-tonal method have begun to give prosody the attention it deserves. This new interest was initially sparked by a number of factors including the growing accessibility of acoustic signal analysis and an emphasis on the communicative function of language (Chun, 1998).

According to Guberina, the prosody acquisition difficulties encountered by both foreign and deaf learners arise from the influence of the sound background of their own system. In the first case, the system is biased by pathological factors, and in the second the learner’s native language system creates a blockage. Language learners will thus perceive the prosody and phonemes of a foreign language through the veil of their own prosodic and phonemic system. This is known as the principle of the ‘phonological sieve’ (Alazard et al., 2010) and it is the basis for the consideration that the learner is “deaf” in the face of target language sounds and prosodic patterns (Germain and Martin, 2000) until otherwise instructed. Since the L1 prosody is so ingrained, learners will produce L2 sentences with L1 prosody superimposed (Alazard et al., 2010) causing misinterpretations or inhibiting comprehension.
The successful acquisition of L2 prosody has a two-fold result; it aids the learner in both being understood, through better pronunciation, and in understanding. Greater understanding has the added value of enabling further language development and improvement as the learner can make intelligent use of their input. At times, the language learner is faced with the incomprehension of their interlocutors, despite excellent vocabulary and a mastery of pronunciation. Correct intonation is vital to comprehension for the reasons illustrated in section 2.3. Creating a greater sensitivity to prosody does not only benefit learners being understood but the learning process also. It is an intuitive fact that correct sentence segmentation as inferred from intonation benefits the learner’s acquisition of new word forms and structures - to learn we must understand. This is the basis, to a certain extent, of Krashen’s Input hypothesis; that the process of learning is enabled by comprehensible input (Pienemann, 2003).

“Humans acquire language in only one way - by understanding messages” Krashen in (Pienemann, 2003, p. 680)

The acquisition of prosody has also been demonstrated to assist reading skills by improving strategies to decode written speech. With reference to Fodor’s Implicit Prosody Hypothesis, Alazard et al suggests; “prosodic structure is mapped onto text to encode the written word-string in silent reading” (Alazard, Astésano & Billières, 2010 :3). The results of a study conducted by the Octogone team at Toulouse’s Mirail University on the reading skills of learners before and after prosodic training indicated a reduction of unaccented syllable duration and the shortening of grammatical and non-grammatical breathing and silent pauses which are indicative of planning. This suggests better fluency after the intonation training. Results from the same study determined that speaker fluency was definitely improved, however, this approach worked best for beginners so it was recommended to be taught at an early stage of learning (Alazard et al., 2010).

Given the advantage pitch perception has to fluency and the difficulty learners face acquiring it, it is clear that intonation training would be of great value. Visualisation in particular may benefit learners by helping them to become aware of these pitch movements; it has already been cited by many as a means of enabling learners to perceive pitch, observe how pitch contours develop as a function of time and illustrate the manner in which parts of a
sentence are related to areas of a contour (Spaal and Hermes, 1993). While a number of speech visualisation programs, outlined as part of the literature review, exist, many fail to give the learner useful feedback by demonstrating what is a pertinent change to intonation and what is not (Germain and Martin, 2000). Pitch-Mod could be introduced to the intonation training program in advance of such programs to ensure pitch awareness and understanding before embarking on more complicated learning tasks.

5.3.2 Other Language Learners

Explicitly creating an awareness of pitch to enable learning would be equally as beneficial to both deaf persons seeking to acquire spoken language and those suffering from specific learning disabilities. Both are faced with communication challenges arising from their production and reception of the prosody of oral speech.

Deaf Learners

With regard to deaf persons, Rosemary O’Halpin cites a number of problems hearing impaired speakers are subject to which impact on their capacity to control prosodic productions. The examples below are somewhat unanimously agreed upon among researchers in the field. Many individuals suffer from respiratory problems such as low lung volume which leads to the number of syllables per breath unit being decreased. Difficulties in co-ordinating the respiratory and laryngeal muscles are also reported, which may result in delayed pauses, problems with rising patterns of intonation and an absence of the characteristic gradual decline of the fundamental frequency across a sentence. Furthermore, failure to create stressed-unstressed distinctions may be due to less proportional shortening or excessive lengthening of phonemes (O’Halpin, 2001).

The overall impression of the speech of the hearing impaired was for a long time described as monotonous however more recent studies have also shown examples of excessive variations in f0, serving no intonational purpose (Spaal and Hermes, 1993). There is a great variation in the performance of the hearing impaired though there is not yet a clear understanding of how or why this might be (O’Halpin, 2001). Pitch, when too low or too high, may be perceived as inappropriate in relation to age or gender and in cases
where the average pitch is inclined to the lower or higher end of the normal dynamic range the introduction of variations can pose difficulties as there is little room left. What was perceived as monotonous speech may in reality be the effect of insufficient variations in pitch; the result is a negative effect on the intelligibility of speech and voice quality (Spaai and Hermes, 1993).

Deaf persons cannot monitor their own productions nor can they access the information being conveyed by their interlocutor’s prosody while lip-reading. The benefits of sensory aids for intonation training are clear with both visualisation as well as tactile systems providing the cues necessary to understanding pitch regulation that cannot be gained from auditory feedback or illustration of the related muscle movements.

An example of the use of visualisation training is a study by O’Halpin which involved the use of “PM Pitch Analyser VII” to train profoundly deaf children in the production of contrastive stress. The subjects were successfully trained to both perceive and produce falling and rising intonation patterns by means of a structured programme using a visual display which displayed the fundamental frequency and intensity contours of both a target and the subject’s production on a split screen (O’Halpin, 2001).

For the profoundly deaf, Pitch-Mod may be about as useful as a piece of paper for illustrating pitch movements. However, for those who maintain even minimal auditory capacity, while sounds may be muffled and faint to the point where speech is not discernible, pitch movements can still be made out. Therefore, people whose hearing is augmented with a hearing aid or are otherwise capable of sufficient sound discrimination but who never learnt oral language may benefit from Pitch-Mod in the same way that foreign language students do - by creating an awareness and an understanding of pitch that facilitates reception and eventual production.

Specific Language Impairment

Similar prosodic difficulties arise in autistic children. Autism has long been assumed to play a role in the atypical expressive prosody found in autistic children that is not seen in those suffering from other disorders, to the extent that it was included in Kanner’s original description of autism (Pépé
et al., 2006). In fact it can be shown that the degree of severity of autism is “strongly, not to say circularly, connected with the degree of language ability” (Péppe et al, 2007: 2). However, prosodic ability in people with autism varies wildly; while some exhibit exaggerated intonation, in others it is deemed monotonous. It is hoped that children with autism can be trained to become more aware of prosody and that this improvement in receptive prosody may also lead to improvements in expressive prosody as receptive prosodic ability is arguably necessary for informing prosodic expressiveness (Péppe et al., 2006).

There are many reasons to associate prosody closely with autism. For example, there is a tendency for autistic people to interpret language literally. Also, theory of mind (the ability to attribute mental states to others) is widely acknowledged as impaired in people with autism (Péppe et al., 2006). As we have seen, mental and emotional states as well as subtler nuances of meaning are conveyed by prosody so it is not inconceivable that prosodic deficiencies may be at the heart of these observed problems. Communication breakdowns can occur due to the pitch produces by autistic persons as linguistic or pragmatic content such as conversational indications may be lost, the speaker may be mis-interpreted as depressed due to their monotonous speech and pitch that is exaggerated can give the impression of patronising or insincere speech (Péppe et al., 2006).

The main elements determined to be atypical in autistic individuals were accent and stress, both the realisation and the place. Further problems are observable in intonation patterns for distinguishing utterance type. In a study testing for prosodic expression and reception administered on an experimental group of varying prosodic ability all individuals were found to have difficulty with at least one aspect of prosody. While many researchers have focused on expressive elements, Pépé considers that there is a likelihood that expressive prosody is to a certain extent caused by a “reduced ability to process prosody or perceive it as meaningful” (Pépé et al., 2007, p. 4). Pépé et al. (2007) determined that receptive and expressive prosodic skills did indeed correlate in function tasks suggesting that targeting receptive skills may indeed be of benefit to expressive skills. Therefore, using visual training to make persons with autism more aware of prosody may lead to improvements in their overall use; Pitch-Mod could be a tool integrated
by clinical speech therapists as an introductory step in intonation training.

5.4 Pitch-Mod in these Contexts

The stimulation of “pure sound perception” in the language learner before performing intonation training is recommended by Spaai and Hermes (1993). In this way users can benefit fully from proposed intonation exercises as an understanding of the relationship between graphical representations and the speech signal has been developed. While many other systems exist that record speakers’ productions and make corrections with feedback, Pitch-Mod effectively fills the gap which they left behind by using visual displays with a focus on perceptual discrimination rather than for directly improving articulation or production. Germain and Martin’s (2000), attempt to address this need by means of an annotation function, while beneficial is still a visual to visual relationship. There is a need to make audio-visual association in order to render the use of visualisation meaningful as is highlighted by de Bot.

“Methods making use of visualisation as an aid in intonation learning wrongly imply, that visualisation by itself is useful in teaching: the tacit assumption that the pupil is an unbiased perceiver of whatever the teacher is presenting leads to many problems, because the perceptual abilities of the pupils in general have not been developed sufficiently and specifically enough for the perception of intonation” de Bot (1981) in (Chun, 1998, 76).

In summary, Pitch-Mod has both functional and research uses. On the one hand it can be used as a teaching device whose primary function is developing an awareness of pitch movements and how they relate to graphical representations, a skill required to effectively exploit the many graphical intonation training programs currently available. On the other hand, Pitch-Mod can be integrated into research laboratories as a tool to support work in speech analysis and synthesis where one possible realisation could be in the performance of perception tests. Far from being a trivial modification and visualisation system, Pitch-Mod opens up significant possibilities for language acquisition for a variety of users in a novel and highly beneficial way.
Chapter 6

Conclusion

6.1 Project Outcome

The aim of this project was to develop an interface for the modelling and modification of pitch contours. By using a combination of the MatLab GUIDE environment to develop a surface graphical display and developing code to implement the necessary functionality of this display the Pitch-Mod interface was produced. This project has achieved its original goal with an intuitive and functional interface that launches and runs smoothly. Efficient and appropriate management of the user’s selections is ensured by guaranteeing that the algorithms at the heart of the program execution were robust, state-of-the-art implementations.

Pitch-mod enable users, both researchers and language learners of different types, to manipulate speech samples by selecting their own pitch contours interactively and graphically. The unique contribution that the Pitch-Mod interface makes, among the many other pitch visualisation and training programs available, is that it enables the development of an awareness and understanding of the link between graphical representations of pitch contours and their acoustic realisation. Through repeated manipulation and practice a user gains insights into how on screen rises and falls correspond to what is heard, and in what way these pitch movements are pertinent to the various functions of intonation. This is of great value for research purposes, in acquiring a new language, or in improving your capacity to communicate in your own language and as such that the programme as developed makes a
useful contribution to the wider TCD work which you referred to earlier on. However, over the course of this project some minor issues and limitations arose which remain to be addressed. These, along with a number of possible future developments of the Pitch-Mod program, are outlined below.

### 6.1.1 Remaining Issues and Limitations

With regard to limitations of Pitch-Mod, its primary restriction is that it is not capable of being run indefinitely appearing to be restricted to the processing power of MatLab which, as a mathematically driven program was not initially designed for the file reading or GUI running employed by Pitch-Mod and it is believed that the program may be suffering from some form of memory leakage.

Finally, within the Pitch-Mod program, modifications to pitch are made by varying the contour around 1, which represents 100% or the original speech pitch. Modifications between 0 and 1 correspond to a pitch lower than the original while between 1 and 2 is considered higher than the input. However, in practical implementations these modifications must be kept within the range of 0.5 to 1.5, as is the case in Pitch-Mod, and it may be necessary to restrict this even further. This is a limitation introduced by the TD-PSOLA algorithm as large modifications of the speech signal by have been found to introduce artefacts and errors in the output (Yang and Sun, 2002). This is not specific to TD-PSOLA, even the most state-of-the-art and recent algorithms will have a region within which they introduce artefacts.

### 6.2 Future Work

A number of optional features were considered towards the end of this project, however due to time constraints their addition was not implemented. These features and constraints remain a possibility for future adaptations of the Pitch-Mod interface.

**Optional Constraints**

One of the additional constraints considered involved the blocking of the DRAW function until a sound file had been loaded. This could involve a
pop-up error message and freezing of other GUI functions until a sound file is selected. Similarly, the imposition of a one contour at a time rule for the DRAW function is desirable functionality which may be introduced. This would be simple to implement with a conditional if-statement testing whether a contour had yet to be entered. Uncertainty surrounding how to throw and catch errors in MatLab prevented the author from introducing this constraint.

Optional Additions

There is wide scope for additional functionality in Pitch-Mod that would be of benefit to target users. Firstly, the interface could easily be altered to handle not only manipulation of pitch but other suprasegmentals such as intensity also. These other speech processing tools could easily be incorporated thanks to the development of Pitch-Mod in MatLab which includes specific matrix handling functions.

With regard to the contour manipulation, the introduction of a RELOAD function would enable users to make use of the information stored during previous uses of the program within the Pitch-Mod interface itself, and continue modifications to that data. A compliment to this would involve a function allowing users to make alterations to the contour displayed on screen, possibly by dragging points along the line into desired positions. These options would mean that users need not start from scratch at each iteration when attempting to generate a sequence of similar contours.

Concluding Remarks

In conclusion, whether or not the above changes are implemented, this project has achieved its purpose of constructing an interface, Pitch-Mod, capable of fulfilling the task of graphically and interactively altering pitch contours. The reality of how users will engage with Pitch-Mod and whether it will indeed enable the foundation an awareness of the relationship between pitch visualisation and its auditory realisation remains to be seen and could be the subject of future empirical study.
Bibliography


Appendix A

```matlab
function varargout = pitch_draw_1(varargin)

end

% The MATLAB code for pitch_draw_1.fig
% PITCH_DRAW_1, by itself, creates a new PITCH_DRAW_1 or raises the existing
% singleton.
% H = PITCH_DRAW_1 returns the handle to a new PITCH_DRAW_1 or the handle to
% the existing singleton.
% PITCH_DRAW_1('CALLBACK', hObject, eventdata, handles,...) calls the local

addpath(genpath(pwd));

% Begin initialization code, code common to all interfaces created in MATLAB

gui_Singleton = 1;

gui_State = struct('gui_Name', 'filename', ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @pitch_draw_1_OpeningFcn, ...
    'gui_OutputFcn', @pitch_draw_1_OutputFcn, ...
    'gui_LayoutFcn', [], ..., ...
    'gui_Callback', []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    varargin{1} = varargin{1:nargin};
    gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% End initialization code
```

43
% --- Executes just before pitch_draw_1 is made visible.
function pitch_draw_1_OpeningFcn(hObject, eventdata, handles, varargin)

%**********Initial Setup**********

% Choose default command line output for pitch_draw_1
handles.output = hObject;
 handles.yvalues = 0;
% Label the plots within the output figures
axes(handles.inputplot);
 axis xy;
 xlabel('Time (s)');
 ylabel('Frequency (Hz)');
 axes(handles.spectro);
 axis xy;
 xlabel('Time (s)');
 ylabel('Frequency (Hz)');
 set(handles.inputplot, 'Ylim', [0.5 1.5]);
 set(handles.inputplot, 'Xlim', [0 1]) ;

% Update handles structure -- saves data
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = pitch_draw_1_OutputFn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
% --- Executes on button press in loader ------ Function which loads the sound file
function loader_callback(hObject, eventdata, handles)
handles.nummod = 0; %number of times modification has occurred
%Read in a file and store information for global use
[filename, pathname] = uigetfile('.wav');
[handles.x, handles.fs] = wavread([pathname '\ ' filename]);

%Thresholds
F0min=20;
F0max=500;

% f0 extraction and voicing decision performed using the Summation of Residual Harmonics
% (SRH) method.
[f0,VUV,~] = SRH_PitchTracking(handles.x, handles.fs, F0min, F0max);
F0median = median(f0(f0>F0min & f0<F0max & VUV==1));

% Glottal closure instants (GCI) are detected using the
% SEDREAMS algorithm.
GCI = SEDREAMS_GCIDetection(handles.x, handles.fs, F0median);
% store to Handle
handles.GCI = GCI;

% Align voicing decisions to GCI locations
VUV_gci = interp1(linspace(1,length(handles.x), length(VUV)), VUV, 1:length(handles.x));
VUV_goi = VUV_gci<0.5 = 0; VUV_goi = VUV_gci>=0.5 = 1;
% store to Handle
handles.VUV_GCI = VUV_goi(GCI);

% *****Plot spectrogram*******
axes(handles.spectro);
plot_spectrogram(handles.x, handles.fs)

%Update handles structure -- saves data for global use
guidata(hObject, handles);
% Executes on button press in draw ---Function which retrieves f0/pitch values from the user

function draw_Callback(hObject, eventdata, handles)

%**********User input set-up**********

inpoints = []; % empty vector of points
num_points = 0; % number of points
yvalues = []; % yvalues stores the y coordinate

hold on

% Loop, registering the points selected by the user

button = 1; % 1 indicates left, 3 indicates right mouse click

while button == 1
    [xi,yi,button] = ginput(3);
    plot(xi,yi,'bo') % plot the points
    num_points = num_points+1;
    inpoints(:,num_points) = [xi yi];
end

% Sort the frequency values with respect to the time points (sort Y with respect to X)
[~, I] = sort(inpoints(1,:));
inpoints = inpoints(:, I);

% Store the x and y coordinate vectors separately
xvalues = inpoints(1,:);
yvalues = inpoints(2,:);
% Actions to be performed on the condition that input is adequate only when
% there is more than one value entered and a sound file has been chosen
if num_points>=2
  if ~isempty(handles.x)
    % Smooth out the plotted curve using interpolation
    t = 1:num_points;
    ts = 1: 0.1: num_points;
    points_inter1 = spline(t, inpoints, ts);
    plot(points_inter1(1,:), points_inter1(2,:), 'b-');
  % Extend the set of the frequency values to the length of the input sound length
  inpoints_interp = interp1(xvalues*length(handles.x), yvalues, length(handles.x), 'spline');
  % Store only those frequency values occurring at glottal closure instants
  handles.GCI;
  yvalues = inpoints_interp(handles.GCI);
  % Store to handles
  handles.yvalues = yvalues;
  end
  end

% Update handles structure -- saves data
  guidata(hObject, handles);

% --- Executes on button press in psola. -- Function which performs
% modification of pitch
function psola_Callback(hObject, eventdata, handles)
% Retrieve global variables for use in the do_f0_modification method

% Sound file
x = handles.x;
fs = handles.fs;
% Vector of pitch values
pscale = handles.yvalues;

% Perform pitch modification by calling method do_f0_modification
handles.y = do_f0_modification(x, fs, pscale, handles.GCI, handles.VUV_GCI);
% Update the number of modifications done
handles.nummod = handles.nummod + 1;

% Update handles structure -- saves data
  guidata(hObject, handles);
% --- Executes on button press in play ---
function play_Callback(hObject, eventdata, handles)
% hObject    handle to play (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

if (handles.nummod == 0)    % no modifications, play unaltered sample
    sound(handles.x, handles.fs);
else
    sound(handles.y, handles.fs);    % otherwise play altered sample
end

% --- Executes on button press in store -- function outputs the pitch
% vector values to a .txt file for future use
function store_Callback(hObject, eventdata, handles)

%Set up file to store into
file_name = 'output.txt';
%open file with accumulative writing permission
fid = fopen(file_name, 'a+');

%Store values of the modified pitch vector if modification has occurred
if(handles.nummod >1)
    freqvals=handles.y.values*150;
    outtext=strcat('Modification Number',num2str(handles.nummod),', ',num2str(freqvals));
else
    outtext = 'No modifications to store';
end
%Print to file :Output.txt which is created in the folder GUI if non-existant
fprintf(fid,'%s\n\r', outtext);
%Close the text file
close(fid);

% --- Executes on button press in reset --- function returns the interface
% to a starting configuration
function reset_Callback(hObject, eventdata, handles)
%clear stored info
axces(handles.inputplot);
clear
guidata(hObject, handles);
Appendix B

function plot_spectogram(x,fs,window,nfft)
    if nargin < 3
        window=256;
    end
    if nargin < 4
        nfft=1024;
    end
    
    % calculate the table of amplitudes
    [B,f,t] = specgram(x,nfft,fs,window,round(window*.8));

    % calculate amplitude 50dB down from maximum
    bmin=max(max(abs(B)))/8000;
    
    % plot top 50dB as image
    imagesc(t,f,20*log10(max(abs(B),bmin)/bmin));
    
    % label plot
    axis xy;
    xlabel('Time (s)');
    ylabel('Frequency (Hz)');
    
    % build and use a grey scale
    lgrays=zeros(100,3);
    for i=1:100
        lgrays(i,:) = 1-i/100;
    end
end
colormap(lgrays);