Audio Engineering Project

Repeated Pitch-shifting for Frequency Discretization as Artistic Effect for Speech Signals

Manuel Planton, BSc

Supervisor: Dr. Robert Höldrich

WS2020

Graz, July 6, 2021

institut für elektronische musik und akustik

Abstract

Pitch-shifting effects have been used for a long time, but there are still new innovations to the employed algorithms for fundamental tone analysis (pitch-tracking) and synthesis of a frequency shifted audio signal (pitch-shifting). This project starts with the research on the contemporary pitch-tracking and pitch-shifting algorithms literature, to develop a frequency discretizing pitch-shifting algorithm for monophonic speech signals. Strategies and techniques of the specific design for the effects implementation are discussed. Finally a pitch-discretization effect is implemented and evaluated. So a monophonic speech signal is discretized according to certain pitch scales. The center of this study is the outcome of the repeated application of this audio effect and which possibilities it offers.

Kurzfassung

Pitch-shifting Effekte werden bereits längere Zeit verwendet, und doch gibt es stets Neuerungen für die verwendeten Algorithmen zur Grundtonanalyse (pitch-tracking) und Synthese eines tonhöhenverschobenen Audiosignals (pitch-shifting). Dieses Projekt geht von einer Literaturrecherche über die aktuellen pitch-tracking und pitch-shifting Algorithmen aus, um einen frequenzdiskretisierenden pitch-shifting Algorithmus für ein monophones Sprachsignal zu Entwickeln. Es werden Strategien und Techniken für das spezifische Design zur Implementierung des Effektes diskutiert. Schließlich wird ein frequenzdiskretisierender Effekt implementiert und evaluiert. Dabei soll ein monophones Sprachsignal in gewissen Tonhöhenskalen diskretisiert werden. Es wird untersucht, was eine mehrmalige Anwendung dieses Effektes zur Folge hat und welche Möglichkeiten sich damit bieten.

Contents

1 Introduction

A *pitch-discretization* effect is an audio signal processing algorithm that changes the pitch of its input signal according to a chosen musical scale. The output signal of this effect then follows the notes of the chosen scale. Sounds between the pitches of the defined scale are discretized in pitch.

This work deals with the employed algorithms and the implementation of a real-time capable pitch-discretization effect for monophonic speech signals. Such a pitch-discretization effect consists of a *pitch-tracking* algorithm that analyzes the pitch of the input signal and of a *pitch-shifting* algorithm that shifts the input signal to the notes of the given musical scale.

The motivation to implement this effect is an artistic experiment which is outlined in figure 1. One stage of the experiment is to apply the developed pitch-discretization on an input signal and sum the output with the original signal. The output of the first stage will consist of two voices and after every stage, the voices will double. The question is how the output sounds after conducting the experiment with different numbers of stages.

Although pitch-shifting and pitch-tracking have been used for a long time, there are still new innovations in pitch-shifting and especially in pitch-tracking algorithms. This work starts with the presentation of the numerous algorithms available.

Then different designs for the implementation of a pitch-discretization effect for monophonic speech signals are developed and discussed. A design for a real-time capable effect for artistic purposes is picked and the used algorithms are explained in detail. The performance of the implemented pitch-tracking and pitch-shifting algorithms and the resulting pitch-discretizing effect are discussed. Finally the artistic experiment is conducted with different stages and the results are evaluated.

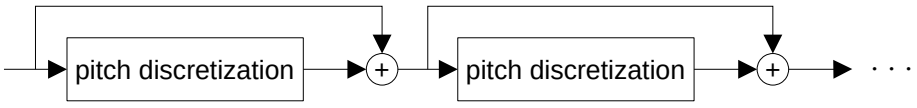

Figure 1 – The artistic experiment.

The developed software for this project is available under `https://git.iem.at/s1061531/reshift` [Pla21b] and `https://github.com/mplanton/pitch_discretization`. Algorithms have been explored in Jupyter notebooks (Python) and the real-time capable audio processing effects have been implemented in Pure Data.

1.1 Pitch-shifting

Pitch-shifting is the procedure of changing the perceived pitch of an audio signal. As this can be achieved in several ways, there are numerous algorithms available for this task. Due to the link of time and frequency $f = 1/t$ it is the dual problem of *time scale modification* (TSM), which performs *time-stretching* or *time-compression*. The goal of TSM is to change the duration of the audio signal independently of its pitch and the goal of pitch-shifting is to change the pitch of the audio signal independently of its duration.

This section is about pitch-shifting algorithms and common artifacts produced by them. Literature that compares these algorithms is shown and current publications are presented.

1.1.1 Algorithms

The easiest way to change a signals pitch is variable speed replay. This can be achieved by playing a record with different speeds. Since this also changes the signals duration, it is not considered pitch-shifting by definition, but it shows the connection between time and frequency.

Different pitch-shifting algorithms produce different artifacts and these can be linked to their processing domains. For this reason I classified the currently available pitch-shifting algorithms into *time-domain algorithms* and *frequency-domain algorithms*. In the following, common artifacts of pitch-shifting algorithms are presented followed by a list of basic pitch-shifting algorithms. The time-domain algorithms work directly on the signal to be processed whereas the frequency-domain algorithms at first have to perform a frequency transform such as the discrete Fourier transform [1] of the signal and finally perform the inverse transform for yielding the time-domain output signal.

Artifacts A prominent side effect of pitch-shifting is *formant shifting* if the pitch-shifting algorithm does not perform *formant preservation* which is sometimes called *timbre preservation*. Such a formant shift occurs for instance if an audio signal like a speech signal is played with a faster speed to shift its pitch up. Then the formants of the signal which represent the characteristic sound of a voice or instrument, are shifted up accordingly. As a result, not only the pitch but also the character of the signal is changed, which is sometimes called the 'Mickey Mouse' effect when formants are shifted upwards. There are different techniques for transient preservation and they are mostly based on a signal model of some kind.

For example *linear predictive coding* (LPC) can be used to model a voice signal to separate the excitation signal from the filtering. Here the excitation signal represents the signal produced from the glottis which is filtered by the vocal tract. Subsequently just the excitation signal is pitch-shifted and then filtered by the vocal tract filter. The result is a higher pitched voice signal with the original formants.

Formant shifting is more prominent at higher shifting intervals and at signals with strong

1. The DFT is mostly implemented with the fast Fourier transform.

Table 1 – Pitch-shifting artifacts of Phase Vocoder techniques (PV), time-domain techniques (TD) and the "Rollers" algorithm [JSA08].

Artifact	**TD**	**PV**	**Rollers**
Phasiness	Yes	No	No
Transient duplications	No	Yes	No
Detuning	Yes	No	Yes
Phase cancellations	No	Yes	No
Tempo modulation	No	Yes	No
Tremolo	Yes	No	Yes
Frequency notches	No	No	Yes
Resonance	No	No	Yes
Stereo field loss	Yes	Yes	No

characteristic formants like voice signals. It is not perceivable at small intervals like a quarter tone or a semitone.

In general "pitch-shifting in the time-domain is characterized by low latency and low quality sounds" and "pitch-shifting in the frequency domain is comparatively characterized by high latency and higher quality than the time domain." [RB19]

Table 1 shows a comparison of artifacts between Phase Vocoder techniques (PV), time-domain techniques (TD) and the "Rollers" algorithm [JSA08] which produces unique artifacts.

Additionally frequency-domain algorithms can also produce *transient smearing* and the PSOLA algorithms can produce static sounds of a synthetic nature by repeating the same pitch period for longer times. To avoid *transient doubling* or *transient smearing*, there are techniques for *transient preservation*. However transient doubling often is an issue at time-domain TSM algorithms but occurs rarely at pitch-shifting algorithms, except when using extreme pitch-shifting ratios.

In the following algorithms lists *(n)* stands for no formant preservation and *(y)* indicates that the algorithm preserves the formants of the signal.

Time-domain Algorithms

- (n) Overlap and Add - OLA (TSM) [DM16]
- (n) Synchronous Overlap and Add - SOLA (TSM) [Zö11, p.201]
- (n) Waveform Similarity Overlap and Add - WSOLA [DM16]
- (n) delay-line modulation [Zö11, p.203]
- (y) Pitch Synchronous Overlap and Add - PSOLA (TSM) with formant preservation [2] [Zö11, p.205]
- (y) Resampling of the excitation signal via LPC [3] [Zö11, p.319]

2. The analysis and the formant preservation is done in the frequency-domain.
3. source filter separation with Linear Predictive Coding

Frequency-domain Algorithms

- Phase Vocoder (PV):
  - (y) PV [Zö11, p.259-63])
  - many improved versions (phase locking, etc.)
- Cepstrum:
  - (y) inverse formant move plus pitch-shifting [Zö11, p.314-17]
  - (y) resampling plus formant move [Zö11, p.317-19]
- (n) Sinusoidal Model (SM) without timbre preservation [Zö11, p.428]
- (y) Sinusoidal Plus Residual Model (SPR) [Zö11, p.434]
- Wavelet based pitch-shifting [Skl06]

1.1.2 Comparative Studies

Driedger and Müller did a review of time-scale modification of music signals in 2016 (as the dual problem to pitch-shifting) and proposed a new TSM algorithm based on harmonic percussive separation [DM16]. In this review, they discussed the artifacts produced by OLA, WSOLA and PV.

In 2019 Rai and Barkana made an analysis of three pitch-shifting algorithms for different musical instruments [RB19]. They compared the "Phase Vocoder using phase locking, PitchshiftOcean, and pitch-shifting via TSM using WSOLA for five musical instruments including guitar, piano, violin, drum, and flute" to "find the most suitable pitch-shifting algorithms for different musical instruments". From their results it can be concluded, that the "Ocean" pitch-shifting algorithm [JH10] causes the least increase in extra noisy components to percussive sounds and TSM WSOLA the most. Furthermore, the "Ocean" algorithm preserves the harmonic component of flute sounds in contrast to the other algorithms.

1.1.3 Current Techniques

Here current techniques are presented starting with the publication date and the name of the paper following with a short description of its content.

2008 Low Latency Audio Pitch Shifting in the Time Domain (Rollers algorithm) [JSA08]
This is a pitch-shifting algorithm for real-time applications that works in the time domain. It is based on frequency-shifting of subbands of a filter bank using IIR filters.

2010 Low Latency Audio Pitch Shifting in the Frequency Domain (Ocean algorithm) [JH10]
This is a pitch-shifting algorithm for real-time applications that works in the frequency domain. The paper also "investigates the problem of latency" in real-time systems. It says that "In a real-time system, the latency of the Phase Vocoder is determined by the size of the Fourier transform. The authors offer a comparison with other algorithms and state that the proposed algorithm "does not rely on an accurate signal analysis."

2010 High Quality Time-domain Pitch Shifting Using PSOLA and Transient Preservation [vdKZZ10]
"An enhanced pitch-shifting system is presented which uses the Pitch Synchronous Overlap Add (PSOLA) technique and a transient detection for processing of monophonic speech or instrument signals." Here the pitch-tracking is done with the YIN [4] algorithm. "A new robust pitch mark positioning algorithm is presented which achieves high quality results and allows the positioning of the pitch marks in a frame-based manner to enable real-time application."

2011 Pitch Shifting by $H\infty$-Optimal Variable Fractional Delay Filters [NHY11]
The authors propose an approximately ten times faster resampling method than traditional methods and a pitch-shifting algorithm without formant preservation tested with a single note guitar tone.

2013 Audio Pitch Shifting Using the Constant-Q Transform [SKS13]
Here a pitch-shifting algorithm is proposed, using the constant-Q transform. "High-quality pitch transpositions with large scaling factors can be achieved without estimating instantaneous frequencies of partials."

2016 TSM Based on Harmonic Percussive Separation [DM16]
This algorithm exploits the benefits of OLA and the Phase Vocoder by separating the input signal into an harmonic and percussive part. They are treated separately and form the output signal. The paper is centered at time scale modification.

2017 Voice Conversion with Pitch Alteration Using Phase Vocoder [LJ17]
The authors propose a voice conversion via spectral envelope transformation and pitch-shifting of the excitation signal. They used a Phase Vocoder approach without instantaneous frequency estimation and an artificial neural network for transforming the signals spectral envelope. "The advantage of the proposed approach is that it avoids explicitly estimating instantaneous F_0."

For pitch-shifting with best sound quality, a frequency-domain algorithm should be used like the improved Phase Vocoder or an approach using the constant-Q transform. In general, time-domain algorithms offer shorter latencies than frequency-domain algorithms.

4. See section 1.2.

1.2 Pitch-tracking

This section is about monophonic pitch-tracking. There is also pitch-tracking of polyphonic signals which is a current field of study and has to be distinguished from monophonic pitch-tracking. Pitch-tracking is the measurement of the perceived pitch of a signal. In the literature, the perceived pitch to be analyzed is mostly used synonymously to the fundamental frequency f_0 of a harmonic signal.

Inharmonic signals like bell sounds or tones produced by FM synthesis can also evoke a pitch perception that is not based on a fundamental frequency. In this case algorithms with a different pitch model should be used. However, in practice the most tonal sounds do have harmonic spectra and a big part of the literature is based on speech analysis, which is a sound mixture of harmonic, noisy and explosive sounds.

In this section a list of pitch-tracking algorithms is presented. Like above, comparative literature and then current publications are summarized briefly.

1.2.1 Algorithms

Pitch-tracking algorithms are a rich field of study. To give an overview, in the following is a list of currently known basic pitch-tracking algorithms. As above, the algorithms are classified into their processing domains. These are *time-domain algorithms*, *frequency-domain algorithms* and *statistical algorithms*.

Time-domain Algorithms

- Event Rate Detection:
  - Zero-crossing rate [Ger03, p.4-5]
  - Peak rate [Ger03, p.5]
  - Slope event rate [Ger03, p.5]
- Average Magnitude Difference Function (AMDF) [Zö11, p.350]
- Average Squared Mean Difference Function (ASMDF) [Zö11, p.350]
- Correlation Based Algorithms:
  - Autocorrelation (ACF) [Zö11, p.347]
  - Weighted Autocorrelation Function (WACF) [PDLCMS01]
  - MPM algorithm using normalized square difference function (NSDF) [MW05]
  - ACF of Linear Predictive Coding (LPC) prediction error [Zö11, p.348]
  - Long-term Prediction (LTP) (ACF for tap-weights update) [Zö11, p.353-60]
  - RAPT (based on normalized cross correlation) [Tal95]
  - PRAAT package default algorithm [Boe93]
  - YIN (based on normalized mean difference function) [CK02] and [Zö11, 350-53]

Frequency-domain Algorithms

- Component frequency ratios [PG79]
- Filter based methods [5] [Ger03, p.10-11]
- Harmonic Peak Detection:
  - Phase Vocoder approach for better frequency resolution [Zö11, p.335-45]
- Harmonic Product Spectrum (HPS) [PDLCMS01]
- Cepstrum [Ger03, p.11]
  - Cepstrum-Biased HPS (CBHPS) [PDLCMS01]
- Summation of Residual Harmonics (SRH) [DA11]
- STRAIGHT [6] [KEF01]

Statistical Algorithms

- Neural Networks [Ger03, p.13]
- Maximum-likelihood [7] (ML) [Smi17, ch. Fundamental Frequency Estimation from Spectral Peaks] and [PDLCMS01]

1.2.2 Comparative Studies

Since 2001 a few studies on comparisons of pitch-tracking algorithms have been conducted. These publications mostly serve a certain field of application. Hence the rankings of the algorithms differ according to the given field.

Cuadra et al. surveyed algorithms for real-time pitch-tracking in 2001 [PDLCMS01]. They say that monophonic pitch-tracking "is a well-researched problem often considered 'solved' in the case of recorded monophonic voices and instruments", but real-time performance is still an issue. Four algorithms suitable for interactive music are presented, which are HPS, ML, CBHPS and WACF.

In 2003 Gerhard made a general comparison of pitch-extraction algorithms [Ger03]. He evaluated four implementations of fundamental frequency estimators with speech and singing voice signals. From his results, it can be concluded, that the YIN algorithm produces solid performance.

Knesebeck and Zölzer compared algorithms for real-time monophonic guitar pitch-tracking in 2010 [vdKZ10]. They concluded, that LTP is the most robust algorithm, but that the YIN algorithm is the most suitable for real-time single note guitar tracking at that time.

In 2011 Drugman and Alwan proposed the SRH algorithm and compared it with six state-of-the-art algorithms in terms of pitch-tracking of speech in noisy environments [DA11]. They show, that SRH performs slightly better than the YIN algorithm in noisy environments.

5. They are described in the frequency-domain but can also be implemented in the time-domain.
6. STRAIGHT could also be classified as a time-domain algorithm, as it uses both frequency- and time-domain analysis.
7. ML could also be classified as frequency-domain algorithm, doing harmonic peak detection.

Babacan et al. made a comparative study of pitch extraction algorithms on a large variety of singing sounds in 2013 [BDd+13]. They found that "PRAAT and RAPT provided the best determination of voicing boundaries. RAPT reached the lowest number of gross pitch errors. YIN achieved the best accuracy." In reverberant conditions "YIN suffered from the strongest degradation, while STRAIGHT was the most robust."

In the literature YIN is often described as a popular and robust algorithm.

1.2.3 Current Techniques

Since 2007 new pitch-tracking algorithms have been proposed and tested against the commonly used ones. In summary, statistical algorithms are trending and outperform the established deterministic approaches in terms of pitch-tracking quality but not necessarily in terms of time complexity.

2007 SWIPE: A Sawtooth Waveform Inspired Pitch Estimator for Speech and Music [CH07]
"SWIPE estimates the pitch as the fundamental frequency of the sawtooth waveform whose spectrum best matches the spectrum of the input signal." The variation SWIPE' outperformed 12 other algorithms; among them were RAPT and YIN.

2014 pYIN: Probabilistic YIN [MD14]
This is the extended probabilistic version of the YIN algorithm. Kim et al. describe it as the "best performing method to date" [KSLB18]. This method produces pitch candidates and uses a Hidden Markov Model (HMM) which is Viterbi-decoded to produce an improved pitch track. It is shown, that pYIN has superior precision to YIN. Furthermore it offers a voiced/unvoiced probability unlike the YIN algorithm.

2018 CREPE: Convolutional Representation for Pitch Estimation [KSLB18]
The authors proposed "a data-driven pitch-tracking algorithm, CREPE, which is based on a deep convolutional neural network that operates directly on the time-domain waveform." They say, it performs equally or better than pYIN.

2019 FCN: Fully-Convolutional Network for Pitch Estimation of Speech Signals [AR19]
This FCN should offer a better approach for real-time usage. The authors say that CREPE is "too computationally heavy" and that their algorithm "outperformed CREPE and SWIPE". It offers no voiced/unvoiced decision (future research) and needs expensive retraining for different settings.

2019 Fully Bayesian Fundamental Frequency Tracking [SNJ+19]
This algorithm is based on the harmonic model and a first-order Markov process. It has been tested against PEFAC, CREPE, YIN, fast NLS and "the proposed fundamental frequency tracking algorithm has superior performance in almost all investigated scenarios, especially in noisy conditions."

2020 ECKF: Extended Complex Kalman Filter [dSiC20]
The ECKF can extract the fundamental frequency from the waveform and yields the amplitude envelope and phase track, along with the pitch track. Parameter selection is not trivial and the algorithm might be good for tracking ornaments [8] of singing voice. It offers comparable performance to YIN and CREPE.

Data-driven approaches like neural networks need large corpora with accurate ground truth for supervised learning. The quality of the corpus for the learning process greatly impacts the quality of the pitch estimation. Recently Gfeller et al. proposed a technique introducing self-supervised learning for pitch estimation in [GFR+20]. The *Fully Bayesian fundamental frequency tracking* approach may produce the qualitatively best results for speech signals to date.

8. Ornaments are melodic and/or rhythmic decorations.

1.3 Pitch-discretization

This section deals with the design of a pitch-discretization effect by using pitch-shifting and pitch-tracking algorithms which were discussed above. First, the general concept of pitch-discretization is investigated followed by an overview of existing pitch-discretization software. Then different designs for the actual implementation are presented. The design for a real-time capable effect for artistic purposes is picked and the implementation of two variations of this design is explained and results are presented.

1.3.1 The Concept

Pitch-discretization is an *adaptive* digital audio effect as Zölzer classified it in [Zö11]. More precisely it is an *auto-adaptive* effect with a feed-forward structure as depicted in figure 2.

Another effect in this category is the compressor. In general, a sound feature is extracted from the input signal and is then scaled. This scaled feature controls a transformation of the input signal to generate the output signal.

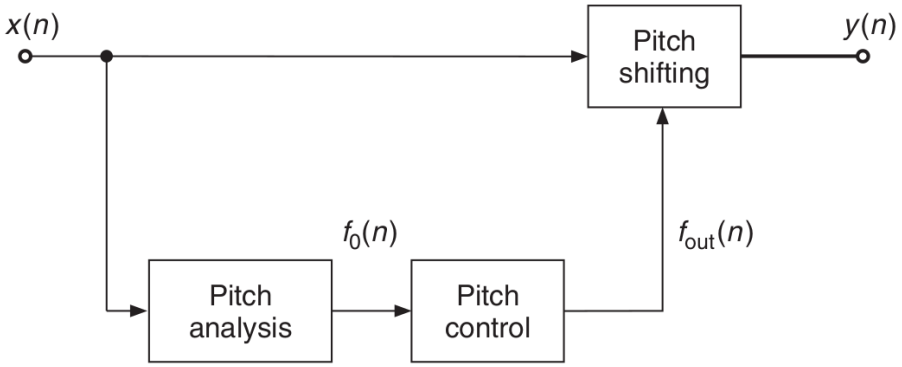

Figure 2 – Pitch-discretization effect [Zö11].

The pitch-discretization effect is also called *auto-tune* (see section 1.3.2 for details) and Zölzer describes it as the following:

> The principle of auto-tuning a musical sound consists in applying a time-varying transposition (or pitch shift), which is controlled by the fundamental frequency of the input signal so that the pitch becomes tuned to a given musical scale (...). It is also called "pitch discretization to temperate scale" (...), even though it would work for any non-temperature scale too. The auto-tune effect modifies the pitch by applying a f_0-dependent pitch shift (...). The output fundamental frequency depends on: (i) the input fundamental frequency, (ii) a non-linear mapping law that applies a discretization of fundamental frequency to the tempered scale (...), and (iii) a musical scale that may change from time to time depending on the music signature and chord changes.

The fundamental frequency $f_0(n)$ (or pitch) of the input signal $x(n)$ is analyzed via a pitch-tracking algorithm from section 1.2. From $f_0(n)$ the target fundamental frequency $f_{out}(n)$ of the output signal $y(n)$ can be computed in the pitch control, which performs the nonlinear frequency scaling. The desired fundamental frequency $f_{out}(n)$ is fed to a pitch-shifting algorithm from section 1.1 to generate $y(n)$ by shifting the frequency of $x(n)$. The pitch-shifting ratio $\rho(n)$ for the pitch-shifting algorithm is computed by

$$\rho(n) = \frac{f_{out}(n)}{f_0(n)}. \tag{1}$$

1.3.2 Existing Software

In this section, popular pitch-discretization software is introduced, divided in proprietary, free and open source software.

Proprietary Popular proprietary software products for automatic pitch correction are *Antares Auto-Tune* [Ant] which is a real-time pitch-discretization software and *Celemony Melodyne* [Cel] which is an offline note editing software. They are extensively used in music production today.

Free

— *GSnap* [Yea]: pitch-correction plugin
— *Graillon* [Aub]: live voice changer
— *KeroVee* [9] [g20]: pitch-correction plugin

Open Source *Autotalent* [Bar] is a real-time pitch-correction LADSPA plugin for monophonic signals with formant preservation. Pitch-tracking is done with an autocorrelation method and pitch-shifting uses a PSOLA algorithm.

TalentedHack [Sal] "is an LV2 port of Tom Baran's Autotalent (...), with added features and improved performance." It uses the MPM algorithm for pitch-tracking instead of an autocorrelation method.

Zita-AT1 [Adr] "is an 'autotuner', normally used to correct the pitch of a voice singing (slightly) out of tune." It uses an autocorrelation method for pitch-tracking and PSOLA for pitch-shifting, like *Autotalent*. "Compared to 'Autotalent' it provides an improved pitch estimation algorithm, and much cleaner resampling. AT1 does not include formant correction, so it should be used to correct small errors only and not to really transpose a song. (...) AT1 can probably be used on some instruments as well, but is primarily designed to cover the vocal range."

9. with simplified version *RoVee*

2 Effect Design

As mentioned above, a pitch-discretization effect includes an algorithm for pitch-tracking and an algorithm for pitch-shifting. Pitch-shifting algorithms avoiding expensive pitch-tracking have no advantage in the design of a pitch-discretization effect, because the accurate pitch estimation has to be done anyway for the overall effect. In the following, three different pitch-discretization algorithm designs are presented.

2.1 Design for Best Sound Quality

When designing a pitch-discretization algorithm for best sound quality, the qualitatively best algorithms for pitch-tracking and pitch-shifting should be used. Most algorithms performances depend on the characteristics of the input signal. In this work the focus is on monophonic speech signals. If the performance in latency and time complexity are neglected and one has to choose the algorithms with the best producing sound quality for pitch-discretization of monophonic speech signals, a system using *fully Bayesian fundamental frequency tracking* [SNJ+19] for pitch-tracking and an *improved Phase Vocoder*, *high quality PSOLA* [vdKZZ10] or *a constant-Q transform approach* [SKS13] for pitch-shifting could produce state of the art sound quality. *Formant correction* might be an issue for high quality effects if bigger discretization intervals are needed.

Since this work focuses on an artistic effect for *real-time usage*, latency and time complexity of the algorithms must be taken into account. The goal is to chain pitch-discretization effects, so the overall latency of the designed algorithm adds up with the number of stages used.

2.2 Design for Best Latency

An interactive musical system may be called real-time capable, if a user performs an action on this system and perceives an instantaneous reaction from it. The acceptable latency of an interactive system may be 20-30ms for the most multimedia and music applications [10] [LK04]. The lower the latency of a signal processing algorithm, the better, since there may be additional latencies in the overall system.

Juillerat et al. pointed out in [JSA08] and [JH10] that the latency of most pitch-shifting algorithms is a trade-off with their sound quality. Most algorithms use block sizes between 2048 and 8192 samples at 44.1kHz. This corresponds to a latency of 46 to 186ms, which is unacceptable for real-time use. The time complexity of the used algorithm adds a few milliseconds to the overall perceived latency. So for real-time effects, algorithms using sample-per-sample processing or small block sizes are preferred.

If sound quality is neglected and the focus of the design is on maximum real-time capability, a system using time-domain algorithms such as *zero-crossing rate* for pitch-tracking and *OLA* with a small block size for pitch-shifting, would have a very small latency.

10. Sometimes it has to be smaller than 5ms.

However, a pitch-correction algorithm like that would suffer from significant artifacts.

2.3 Design for a Real-time Artistic Purpose

The concept: Artifacts as artistic effects The sound quality of an audio processing system can be defined as the absence of unwanted artifacts in its output signal. If audio effects are seen as tools, the tools with the best sound quality are the most valuable. Therefore the effect producing the least amount of artifacts is the best tool for the given purpose.

On the other hand, music production, music performance and sound design are also art forms, which demand artistic tools. Using signal processing as an artistic tool leads to a different perception of artifacts. From this point of view, artifacts are not phenomena, which have to be avoided at any cost, but rather could be used for artistic purposes.

Audio signal processing techniques produce inherent artifacts which always have been used for their artistic value. For instance preamps of mixers have been used for overdriving input signals and magnetic tape recording for tape compression and saturation. Effect units get their characteristic sound also from their artifacts. Hence, when designing an audio effect, it is not the only way to reduce the artifacts of the available algorithms to their minimum, and just benchmark their performance in a technical way. With the concept of artifacts as artistic effects, the effects designer can incorporate the selection of the sound of the artifacts in the design process. This leads to designing a desired sound of the audio effect, which is sound design on a signal processing engineering level.

The design In order to implement a real-time effect, time-domain algorithms may be preferred over frequency-domain algorithms. Time-domain algorithms avoid bigger block sizes and do not depend on a time-frequency transform which might add additional latency.

The YIN algorithm [CK02] has been chosen for the design of the artistic real-time effect due to its wide usage, good performance and accuracy. Furthermore, it is a time-domain algorithm capable of using a small block size dependent of the minimum trackable frequency.

Two versions of this design regarding the employed pitch-shifting algorithm are presented. The first version uses *pitch-shifting by delay line modulation* [Zö11, p.203] which has been chosen because of its comparable good quality given its low time-complexity and simple implementation. The second version uses the *Rollers* [JSA08] algorithm, which has been chosen because it is a more recent time-domain algorithm that does not use signal blocks. The output signal can be computed sample-per-sample and hence it causes a very small latency. Furthermore, it produces unique artifacts (see figure 1) which seemed artistically interesting. Both of these pitch-shifting algorithms can be implemented entirely in the time-domain and produce minimum latencies compared to other pitch-shifting algorithms.

2.4 Implementation

To conduct the artistic experiment, a concrete implementation of pitch-discretization is needed. The *design for a real-time artistic purpose* has been chosen as described above. It uses *YIN* for pitch-tracking and for pitch-shifting one version uses *delay line modulation*, which can also be described as granular synthesis, and the other version uses the *Rollers* algorithm.

Figure 3 shows the structure of the implementation.

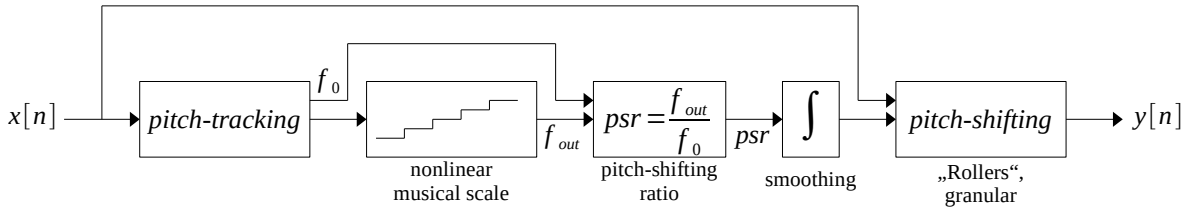

Figure 3 – Real-time pitch-discretization effect implementation.

An input signal $x[n]$ is fed to the pitch-tracking algorithm (YIN) and to the pitch-shifting algorithm (either Rollers or granular). The pitch-tracker analyzes the fundamental frequency f_0 of $x[n]$. Via a *nonlinear musical scale* like a major scale or a chromatic scale, f_0 is discretized and produces the target fundamental frequency f_{out}. Then the pitch-shifting ratio is calculated via $psr = \frac{f_{out}}{f_0}$ to control the pitch-shifting algorithm. The psr is smoothed to avoid too hard jumps between the tones of the musical scale. Finally, the pitch-shifter forms the output signal $y[n]$. This is a pitch discretized version of $x[n]$.

3 Methods

In this section the employed algorithms are elucidated by looking at the signal processing stages of the algorithms. [11]

3.1 YIN Pitch-tracking

The YIN pitch-tracking algorithm [CK02] is named after the Taoist philosophy of "yin" and "yang". It is described originally in 6 steps, but in practice, the first two or three steps can be combined in one processing stage and the last step is omitted [12]. Every new step reduces error rates from 10% gross error (step 1) to 0.77% gross error (step 5) [13].

The calculations in the used YIN steps are:

1. Autocorrelation function (ACF):

$$r_t(\tau) = \sum_{j=t+1}^{t+W} x_j \, x_{j+\tau}$$

2. Difference function:

$$d_t(\tau) = \sum_{j=1}^{W} (x_j - x_{j+\tau})^2$$

3. Cumulative mean normalized difference function (CMNDF):

$$d'_t(\tau) = \begin{cases} 1 & if \quad \tau = 0 \\ \frac{d_t(\tau)}{\frac{1}{\tau}\sum_{j=1}^{\tau} d_t(j)} \end{cases}$$

4. Get the first minimum of the CMNDF under a threshold.
5. Parabolic interpolation at the minimum

To illustrate these steps, they are performed on a speech signal of a male speaker sampled at 44.1 kHz and presented below (see figure 4 to 6). The used signal block is 882 samples or 20ms long.

Figure 4 a) shows the input signal $x[n]$ and the first YIN step (ACF) can be seen in b). By looking at $x[n]$ the quasi-periodic structure of the time signal is already visible and can be estimated roughly. The signals period may be somewhere around 300 to 350 samples. The autocorrelation function (ACF) has maxima at integer multiples of the period of $x[n]$ including zero. Steps 2 and 3 are pictured in c) and d). The difference function has minima at integer multiples of the period of $x[n]$ including zero. The same is true for the CMNDF but excluding zero, so the global minimum is likely to be at the period of $x[n]$. Furthermore the CMNDF is amplitude independent unlike the previous steps.

11. Excluding *pitch-shifting by delay-line modulation* due to its simplicity.
12. See the C++ implementation as a Vamp-plugin of pYIN from Matthias Mauch at `https://code.soundsoftware.ac.uk/projects/pyin/files` [MD14].
13. The errors are defined in [CK02] sec. III.

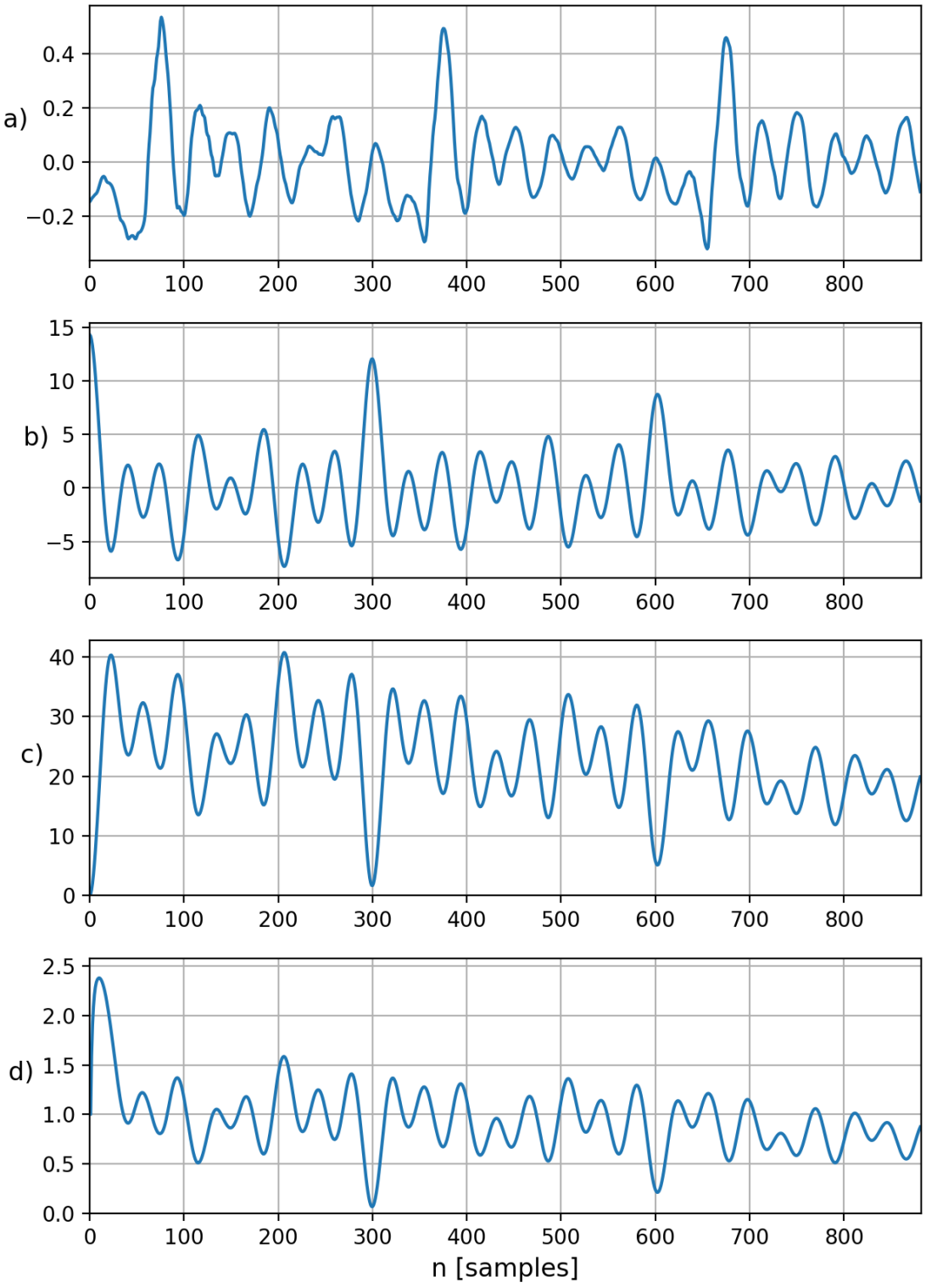

Figure 4 – YIN steps 1 to 3: a) male speech $x[n]$, b) autocorrelation function of $x[n]$, c) difference function of $x[n]$, d) cumulative mean normalized difference function of $x[n]$.

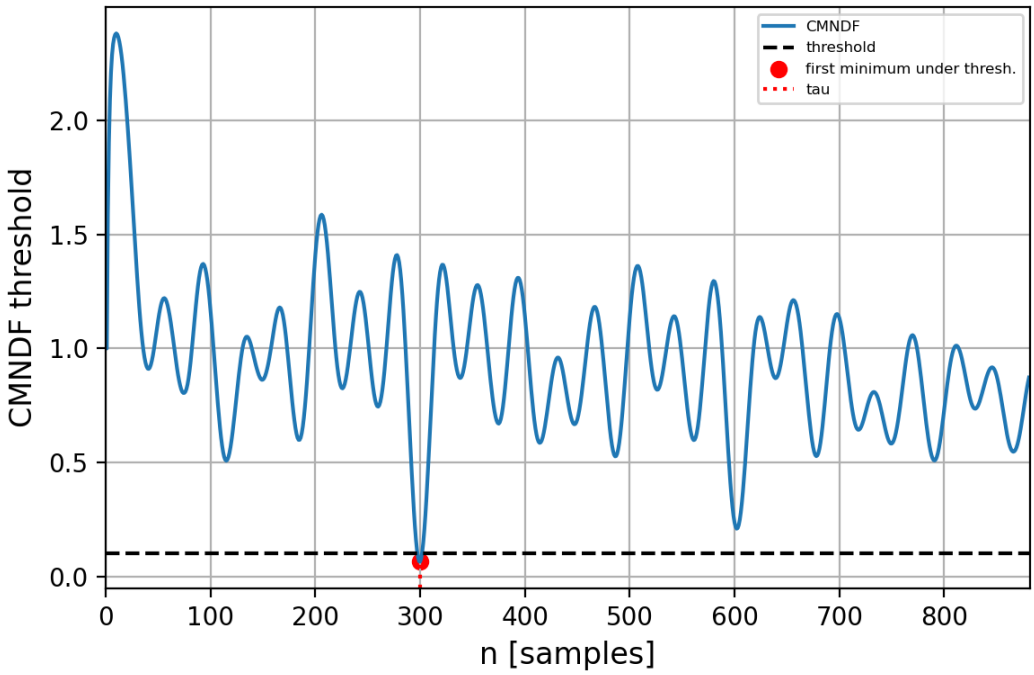

Figure 5 – YIN step 4: First minimum under the threshold.

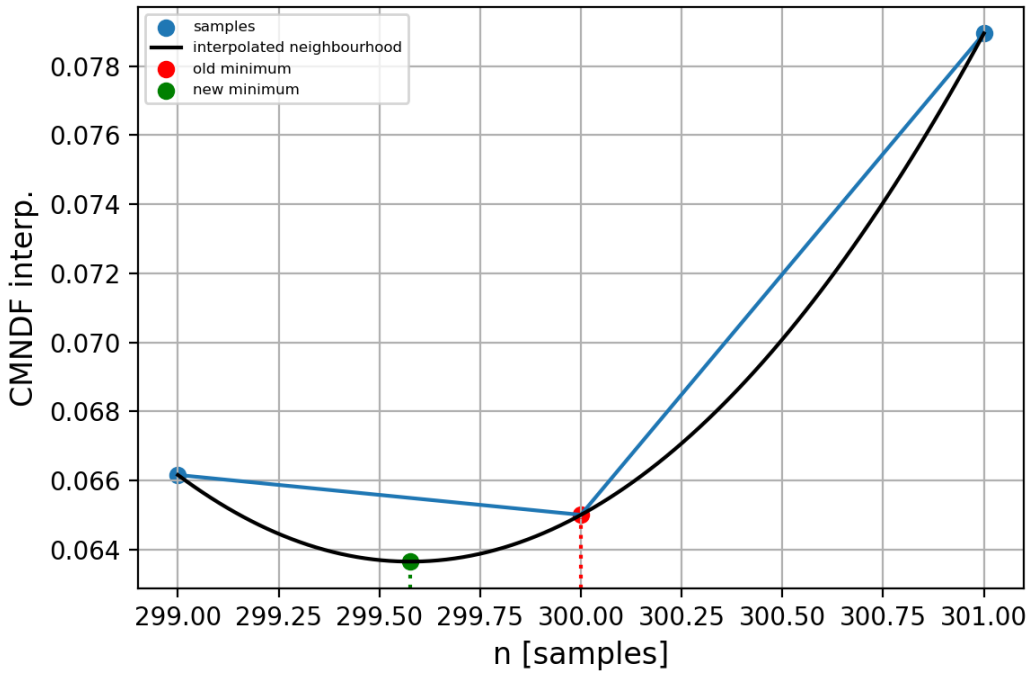

Figure 6 – YIN step 5 parabolic interpolation.

Step 4 is pictured in figure 5 and consists of determining the first minimum of the CMNDF under a threshold. A common value for this threshold is 0.1, like in this example. If no minimum under the threshold is found, the analysis block is marked as unvoiced. At this stage it is very plausible that the period of $x[n]$ lies at around 300 samples.

Figure 6 shows step 5 of the YIN algorithm. This step mostly reduces the fine error by parabolic interpolation. A parabola is fitted through the sample at the minimum and through its neighbors. Then a new refined minimum is determined by calculating the minimum of this parabola, to get sub-sample accuracy.

The old minimum lies at sample 300 which corresponds to 147Hz and the new refined minimum lies at 299.58 samples which corresponds to 147.21Hz.

Performance To show the performance of YIN in practice, the YIN algorithm has been applied to 5 seconds of male speech saying the words "and if to live, The fewer men, the greater share of honor." [Sha02]. Figure 7 shows the spectrogram of the speech signal with the tracked fundamental frequency f_0. Figure 8 shows solely the tracked fundamental frequency f_0 of the speech signal.

As before, the speech signal is sampled at 44.1kHz. The YIN algorithm is set to track a minimum frequency $f_{min} = 100\,\text{Hz}$. This limitation results from the used block size N to calculate the ACF, which needs at least $N \geq 2T_{max}f_s = \frac{2f_s}{f_{min}}$ samples for correct pitch-tracking.

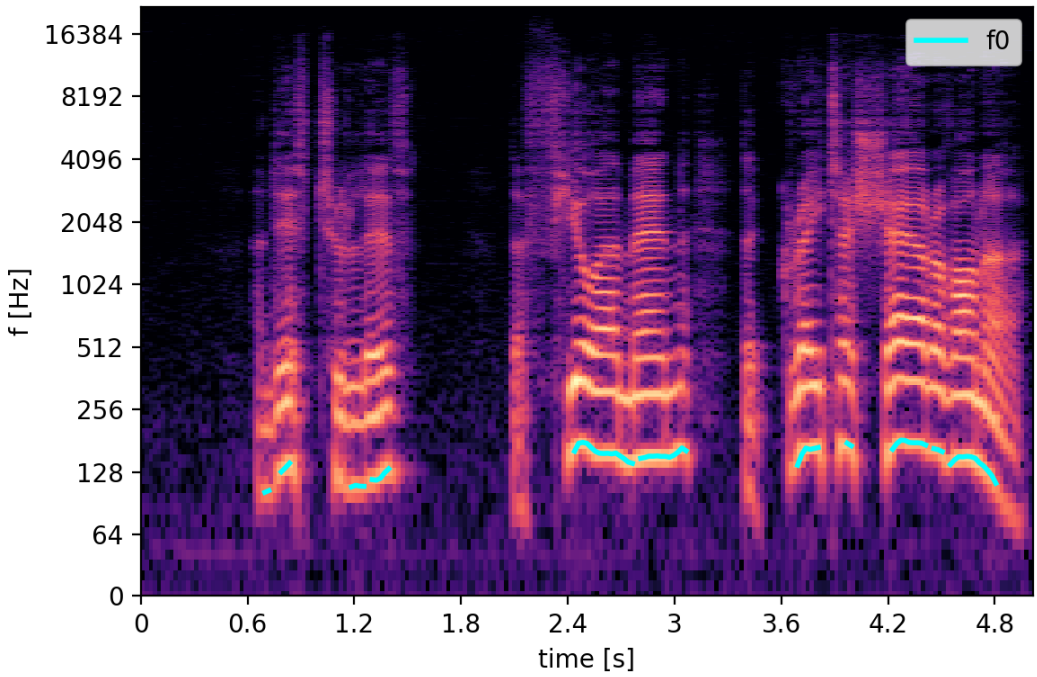

Figure 7 – YIN pitch-tracking: spectrogram of male speech with tracked fundamental frequency f_0 ($f_{min} = 100\,\text{Hz}$, 4096 point FFT with a hop size of 1024 samples).

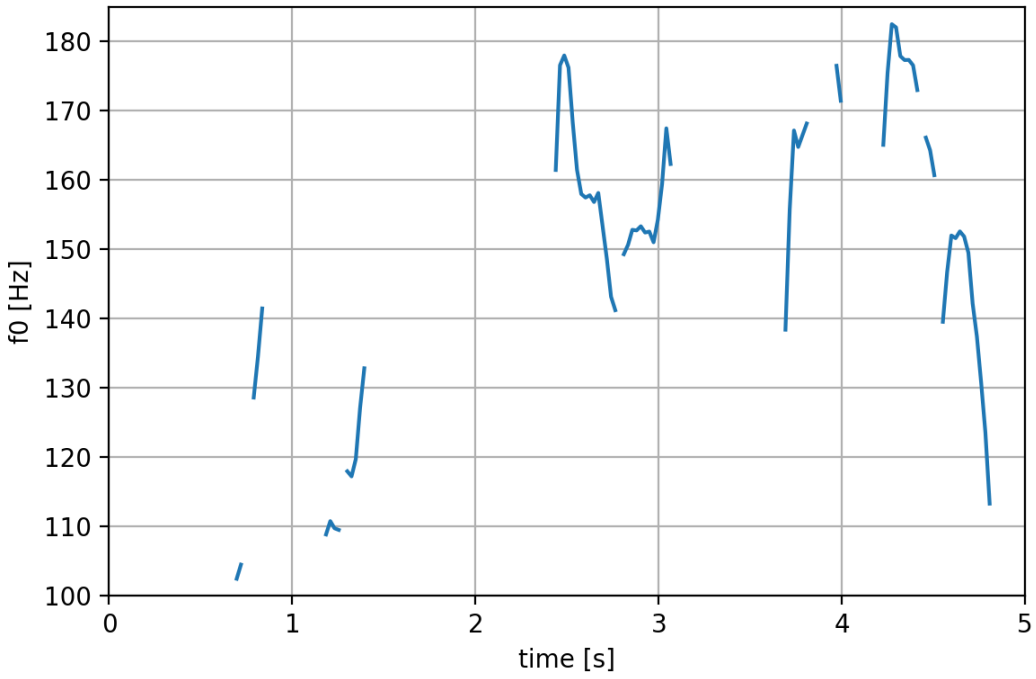

Figure 8 – YIN pitch-tracking ($f_{min} = 100$ Hz, hop size is 1024 samples).

By visually inspecting figure 7 and 8 it can be claimed, that the fundamental frequency of the speech signal is tracked in a reasonable accuracy most of the time, when the fundamental frequency is above 100Hz. In the cases, when $f_0 < f_{min} = 100$ Hz, the algorithm classifies the blocks as unvoiced. Only the voiced f_0 estimations are shown. Unvoiced sections are not plotted in the diagrams.

3.2 Pitch-shifting by Delay-line Modulation

A simple and efficient method for pitch-shifting without transient preservation is *delay-line modulation* [Zö11, p. 203] which also can be called *granular* pitch-shifting [Puc07, pp. 202, 208]. The algorithm is depicted in figure 9.

An input signal is fed to two delay-lines. Via the sawtooth control signals $a(n)$ and $b(n)$ which are 180° out of phase the read-position from these delay-lines is modulated. [14] The outputs are cross-faded, so there are no audible clicks when "jumping" back in the read position, and they are summed together to form the pitch-shifted output signal.

By controlling the frequency of $a(n)$ and $b(n)$ simultaneously, the pitch of the output signal is changed. For the implementation of fractional delay-lines see [Smi21, ch. Delay-Line and Signal Interpolation] and [Zö11, p. 73]. In signal processing systems like Pure Data such fractional delay-lines are already implemented.

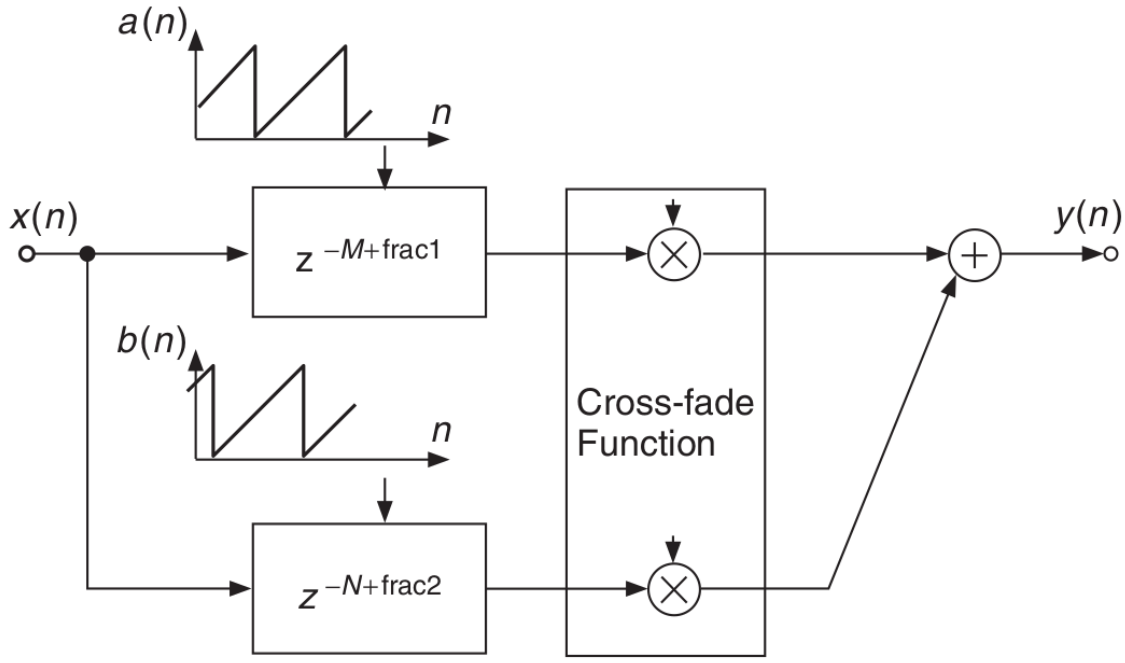

Figure 9 – Delay-line modulation [Zö11].

14. In the early days of electronic music, mechanical tape machines with rotating tape heads had been used in that manner.

3.3 Pitch-shifting with the Rollers Algorithm

Another pitch-shifting algorithm that can work strictly in the time-domain is the "Rollers" algorithm [JSA08]. The algorithm uses a large IIR filter bank and and is based on frequency-shifting of the subband signals. Figure 10 shows the overall concept of the algorithm.

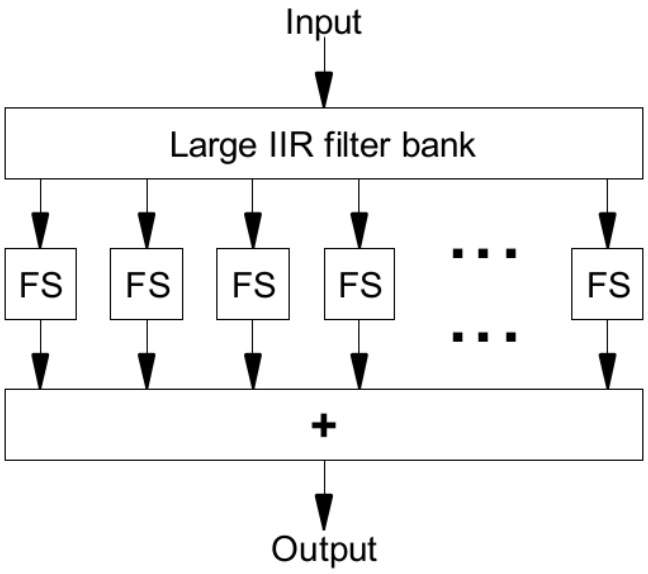

Figure 10 – Rollers concept [JSA08].

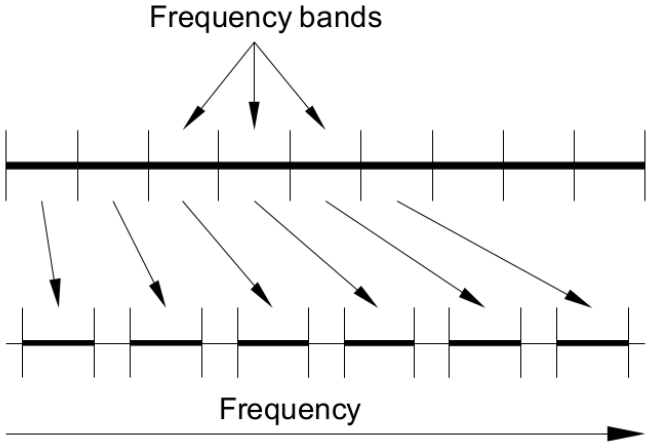

Figure 11 – Rollers pitch-shifting by frequency-shifting of subbands [JSA08].

It is important to distinguish between *pitch-shifting* and *frequency-shifting*. While pitch-shifting scales every frequency in the spectrum of a sound by a constant, frequency-shifting adds a constant to every frequency in the spectrum. So pitch-shifting preserves the harmonic relationship of a sound whereas frequency-shifting results in an inharmonic

spectrum. Figure 11 shows the way, frequency bands of the input signal get shifted by the Rollers algorithm.

Every subband is considered narrow enough to use frequency-shifting for implementing pitch-shifting. If a narrow band signal similar to a sine is frequency-shifted, there are no other frequencies in the spectrum to form an inharmonic relationship. The necessary frequency-shift for every band is given by $f_{shift} = f_c \cdot k - f_c$. Where f_c is the center frequency of the band and k is the shifting factor.

In the following the implementation of the Rollers pitch-shifting algorithm is explained.

Filter bank The filter bank for the Rollers algorithm is a constant-Q IIR filter bank consisting of at least 100 logarithmically spaced second order Butterworth bandpass filters. Figure 12 shows the magnitude response of an exemplary filter bank with 10 filters on a logarithmic frequency axis.

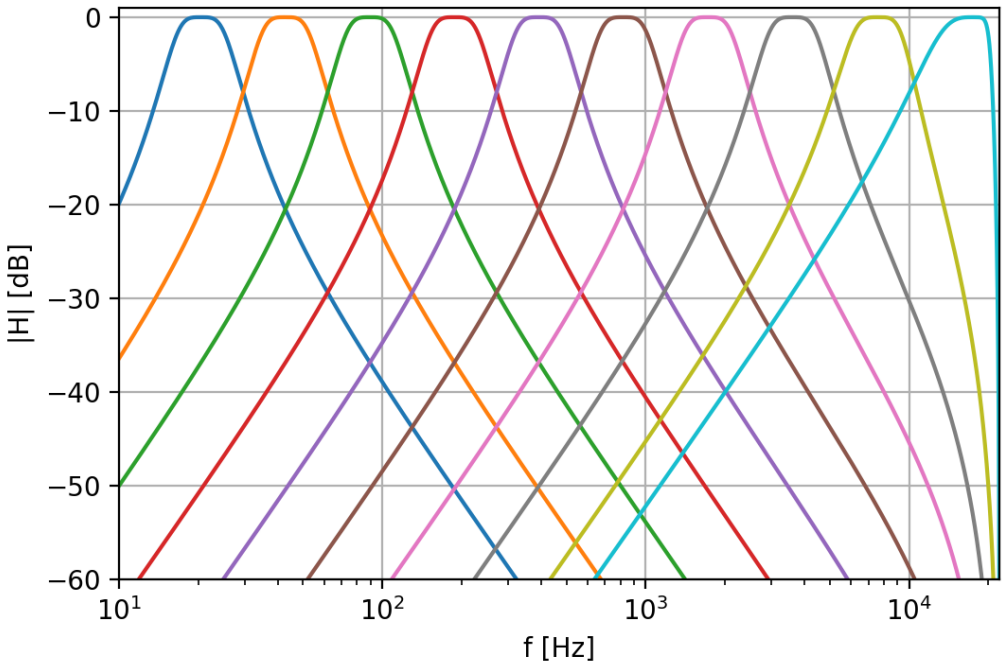

Figure 12 – Logarithmically spaced second order Butterworth filter bank magnitude response on a logarithmic frequency axis.

Figure 13 shows the same filter bank but on a linear frequency axis. The filters for low subbands have very small bandwidths leading to high resonance. This is an issue that results in artifacts which are discussed later in this section.

Now, that the input signal is split into narrow subbands, the frequency-shift is applied in every band. The easiest way to shift frequency spectra that comes to mind is to use amplitude modulation (AM). AM has the disadvantage that the resulting spectrum consists

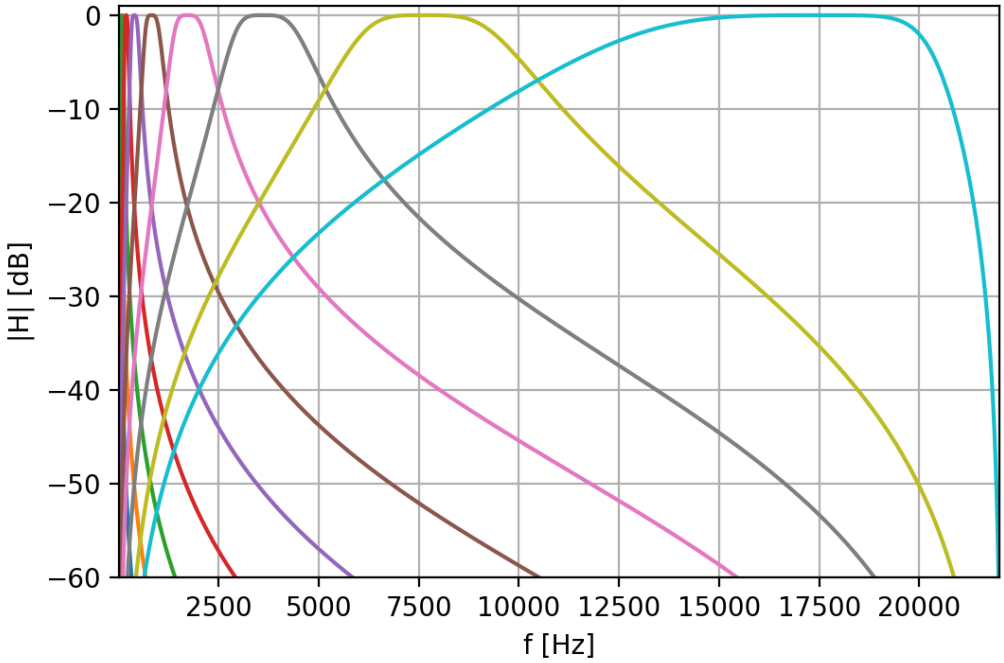

Figure 13 – Logarithmically spaced second order Butterworth filter bank magnitude response on a linear frequency axis.

of the shifted spectrum (upper sideband) as desired, but additionally of the carrier signal and the lower sideband.

To just shift the spectrum on the frequency axis, *single sideband modulation* can be used.

Single sideband modulation (SSB) To implement frequency-shifting in the Rollers subbands, SSB has been used. As mentioned before, this modulation technique has the advantage of just shifting the modulated signal in frequency and does not add anything to it.

Here is an overview of the signal processing stages of SSB:

— Calculate the *analytic signal* by using the *Hilbert transform*.
— Modulate the complex carrier signal.
— Take the real part of the modulated signal.

The continuous time Hilbert transform is given by the convolution $y(t) = x(t) * \frac{1}{\pi t}$ [Smi17, sec. Hilbert Transform], so its Fourier transform is $H_H(j\omega) = -j \cdot sgn(\omega)$ which is diagrammed in figure 14.

This corresponds to a phase shift of $\pi/2$ or $+90°$ for negative frequencies and of $-\pi/2$ or $-90°$ for positive frequencies. A way to approximate the Hilbert transform is to use parallel second order allpass filters like used in [Puc07, p. 259]. This has also been used in the implementation.

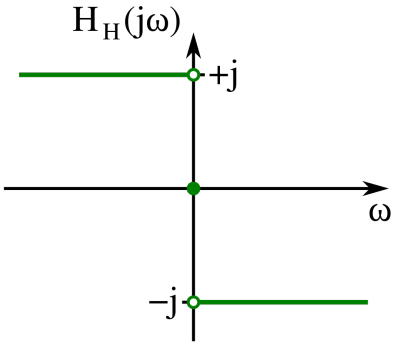

Figure 14 – Continuous time Fourier transform of the Hilbert transform [Stu21].

The *analytic signal* [15] $x_a[n]$ [16] is a complex signal whose imaginary part is formed from the real part via the Hilbert transform $H(\cdot)$

$$x_a[n] = x[n] + jH(x[n]). \tag{2}$$

A full example of frequency-shifting using SSB is pictured in figure 15. In the example, the full spectra of the signals (positive and negative frequencies) are shown to picture the symmetry or asymmetry of the spectra.

The spectrum of the input signal or base band signal $x[n]$ is shown in a). It is a harmonic signal with a fundamental frequency of 1000Hz consisting of three partials. Since every real valued signal has a symmetric spectrum around 0Hz, the spectrum of the real signal $x[n]$ is also symmetric. This is the reason why AM produces a lower sideband in the positive frequencies. The lower sideband gets shifted up to the positive frequency range by the modulation.

To avoid this issue, the analytic signal $x_a[n]$ is formed as in equation (2). As can be seen in b) the spectrum is not symmetric anymore because the spectrum of the analytic signal is single sided. The frequency-shift of the analytic signal is done with a complex carrier signal

$$x_{SSBa}[n] = x_a[n] \cdot e^{j2\pi f_{shift} n/f_s}$$

and the spectrum is pictured in c). The shifting-frequency $f_{shift} = 9000$Hz and f_s is the sampling frequency of the signals.

Finally the real part of the modulated signal is taken $x_{SSB}[n] = Re\{x_{SSBa}[n]\}$. Now the spectrum is shifted and again symmetric around the y-axis. As desired $x_{SSB}[n]$ is the frequency-shifted signal to $x[n]$. The signals has been shifted from 1kHz to 10kHz but the partials do not form a harmonic relationship anymore.

15. From the analytic signal, the amplitude envelope of $x[n]$ can be obtained by the magnitude $|x_a[n]|$ and the instantaneous frequency of $x[n]$ can be obtained by differentiating the instantaneous phase of $x_a[n]$ with respect to time.

16. This is the discrete analytic signal using the discrete Hilbert transform.

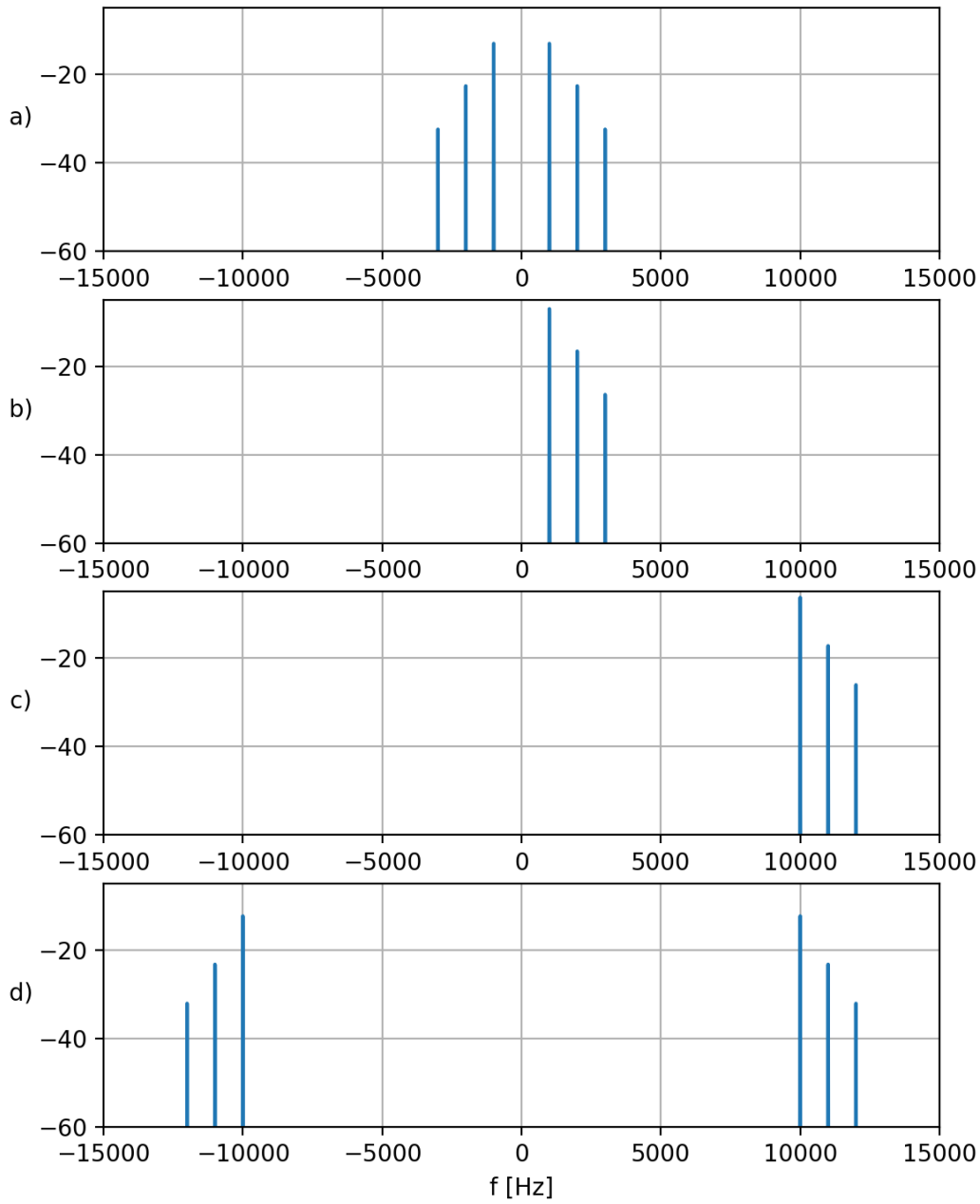

Figure 15 – SSB frequency-shifting: a) $x[n]$ base band signal, b) $x_a[n]$ analytic signal, c) $x_{SSBa}[n]$ modulated analytic signal, d) $x_{SSB}[n]$ single sideband modulated signal.

By applying SSB to the subbands of a filter bank with 200 bands, good results can be achieved. The high resonances in the lower bands lead to a downward chirp artifact and the imperfectly narrow bands lead to an overlap of shifted frequencies, which results in a detuning/tremolo artifact (see figure 1). The artifact produced by the overlap of shifted frequencies can be reduced by leaving bigger spaces between the subbands, leading to notches in the output signals spectrum. For further discussion about the artifacts of the Rollers algorithm see [JSA08].

4 Results

As described above with the presented algorithms, two real-time pitch-discretization effects have been implemented in Pure Data as pictured in figure 3. One uses a granular [17] pitch-shifting algorithm (see section 3.2) and the other one uses the Rollers algorithm (see section 3.3). Both employ the YIN algorithm for pitch-tracking (see 3.1).

For presenting the results, male speech from the same recording as used above has been processed by the developed effects. Figure 16 shows the spectrogram of the original recording.

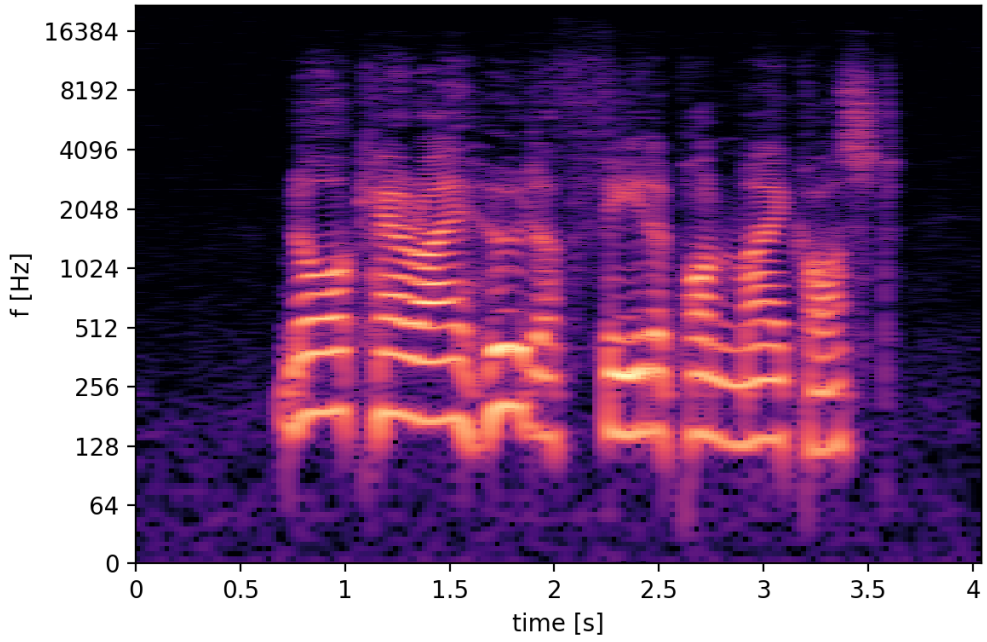

Figure 16 – Spectrogram of original male speech.

The speaker reads the sentence "Nor care I who doth feed upon my cost;" from [Sha02]. A spectrogram of the processed speech signal by the granular pitch-discretization effect is depicted in figure 17 and another spectrogram of the processed speech using the Rollers pitch-discretization effect is depicted in figure 18.

All spectrograms are created with a 4096-point FFT and a hop size of 1024 samples. Both effects use a chromatic scale and no smoothing to show discrete frequency jumps in the partials. The granular pitch-shifter was set to a grain size of 20ms and the Rollers algorithm was set to use a filter bank with 200 bands.

Pronounced frequency jumps are visible in both spectrograms at 1.4s and around 2.9s. Comparing the original spectrogram to the processed ones, it is noticeable that smooth frequency transitions of partials are flattened horizontally. This corresponds to sounds with fixed pitch, which is exactly what has been attempted. Furthermore, the downward

17. pitch-shifting by delay-line modulation

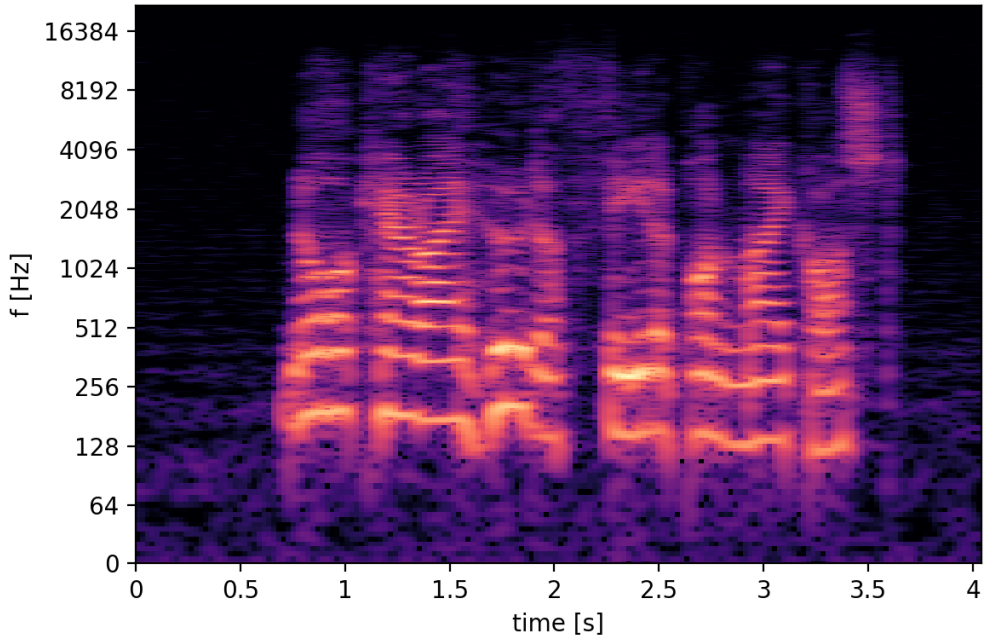

Figure 17 – Male speech discretized with the granular algorithm.

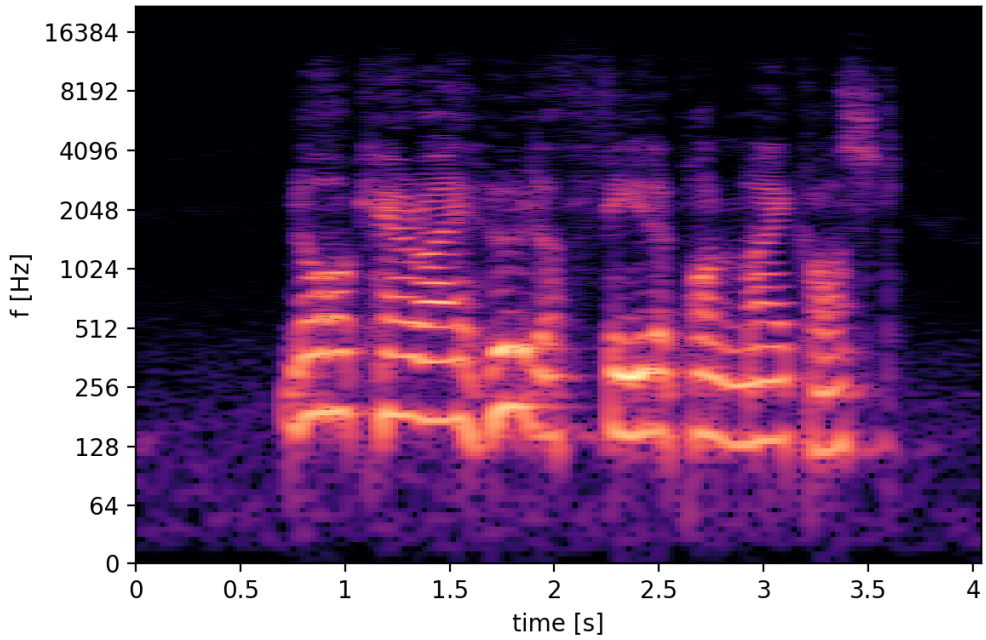

Figure 18 – Male speech discretized with the Rollers algorithm.

chirp artifact caused by the filter resonances of the Rollers algorithm is visible in figure 18 especially in the low frequency range. These described phenomena are clearly audible in the processed audio signal.

4.1 Artistic Experiment

The motivation for this project was the artistic experiment [18] outlined in figure 1. A speech signal is discretized in pitch and summed with the original signal. This is considered as one stage of the experiment. For the pitch-discretization current real-time capable algorithms should be used to develop a real-time effect.

The question was, how it sounds when such processing is applied repeatedly to the speech signal. After one stage, the signal consists of two voices and after stage two, the signal has four voices and so on, so the audio becomes polyphonic. The pitch-tracking algorithm is not designed to track polyphonic pitch, but monophonic pitch, so it will behave differently than in the monophonic case. Another question was, how the voices will spread out. Will they accumulate at one pitch or will they spread over some spectrum? And if they spread, how are they distributed?

Figure 19 to 24 show the spectrograms of the results from the artistic experiment after 1, 2 and 4 stages for the two developed pitch-discretization algorithms. The input signal was the same male speech signal as above. The original spectrogram is depicted in figure 16. As above all spectrograms are created with a 4096-point FFT and a hop size of 1024 samples. Both effects use a chromatic scale and no smoothing to show discrete frequency jumps in the partials. The granular pitch-shifter was set to a grain size of 20ms and the Rollers algorithm was set to use 200 bands.

After stage 1 (fig. 19 and 20) two voices can be heard; the original that behaves with natural smooth pitch slides and the discretized voice which jumps between the notes of the chromatic scale. The granular pitch-discretization has few artifacts but the Rollers algorithm produces resonances at low frequencies which ring at and boost the low frequency range. This is visible in the spectrogram and it is even more clear after stage 2 in figure 22 and after stage 4 in figure 24. From stage 2 onwards the downward chirp artifact is clearly audible.

The granular effect stages sound more neutral than the Rollers stages. Instead of a bass boost it comes to a more evenly spread of the voices. This is a sign that the pitch-tracking algorithm jumps around between the generated voices.

As mentioned above, the developed software used in this project is available under `https://git.iem.at/s1061531/reshift` [Pla21b]. This repository includes the used pitch-shifting and pitch-tracking algorithms and the implemented pitch-discretization effects. Furthermore, it includes the artistic experiments described above and Jupyter notebooks for the employed algorithms and signal processing techniques.

18. A demonstration video of this artistic experiment is available under `https://www.youtube.com/watch?v=o4CzSDUZVtY` [Pla21a].

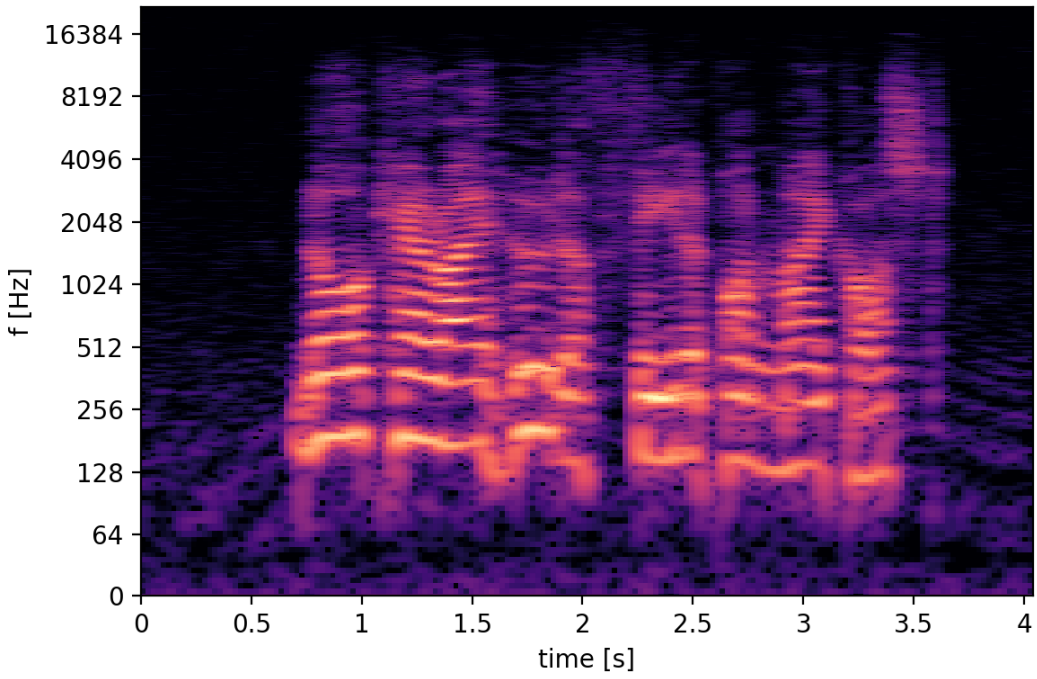

Figure 19 – Stage 1 with granular pitch-discretization.

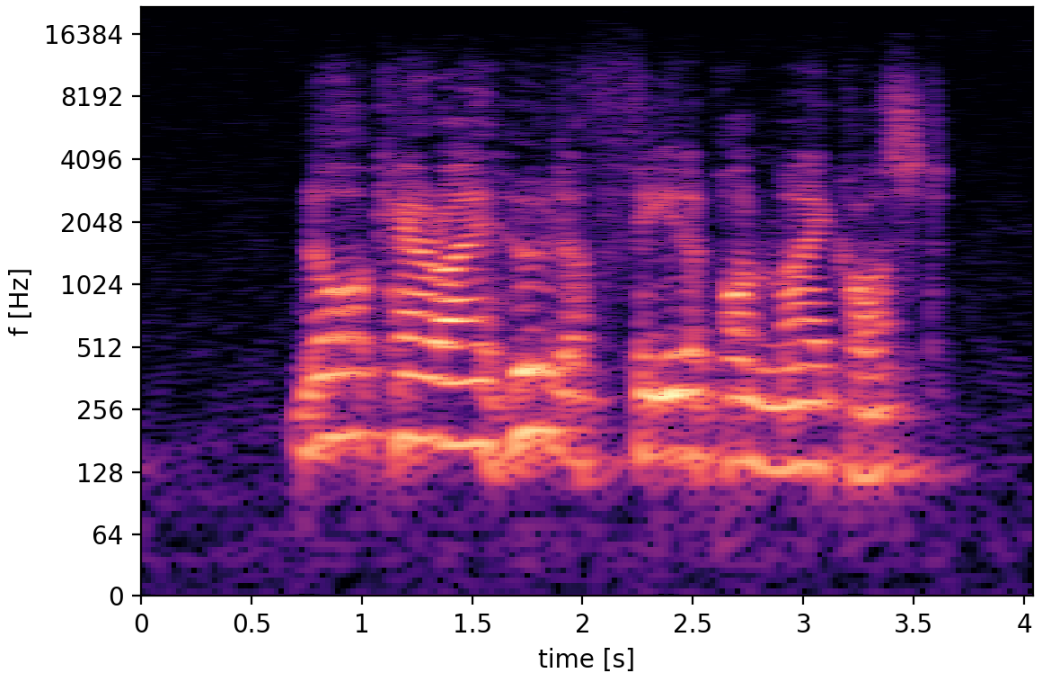

Figure 20 – Stage 1 with Rollers pitch-discretization.

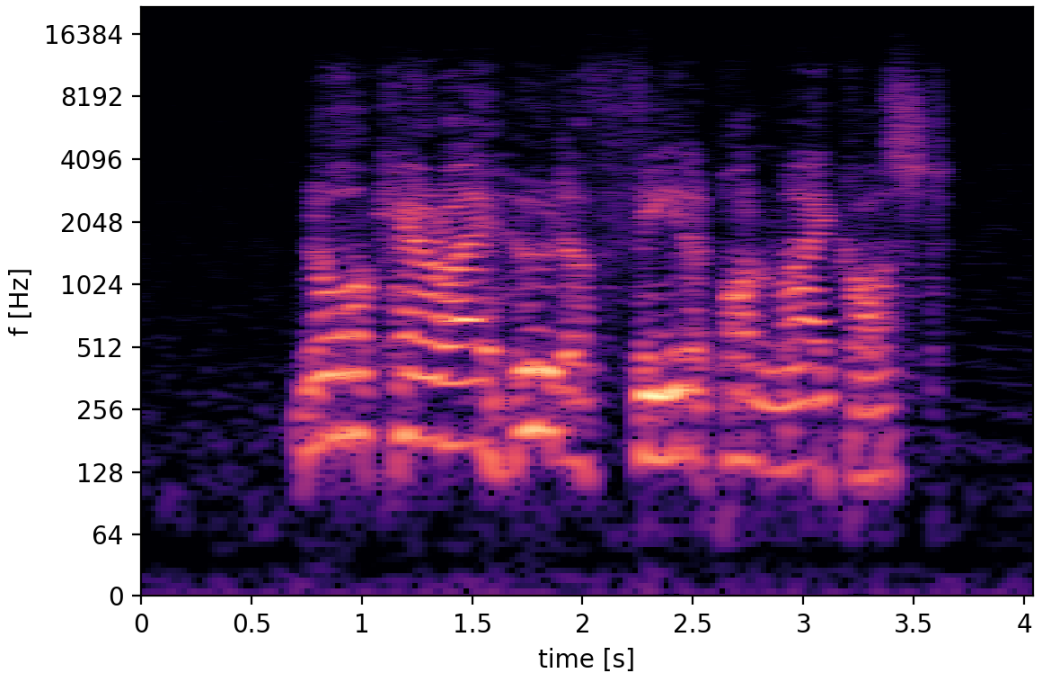

Figure 21 – Stage 2 with granular pitch-discretization.

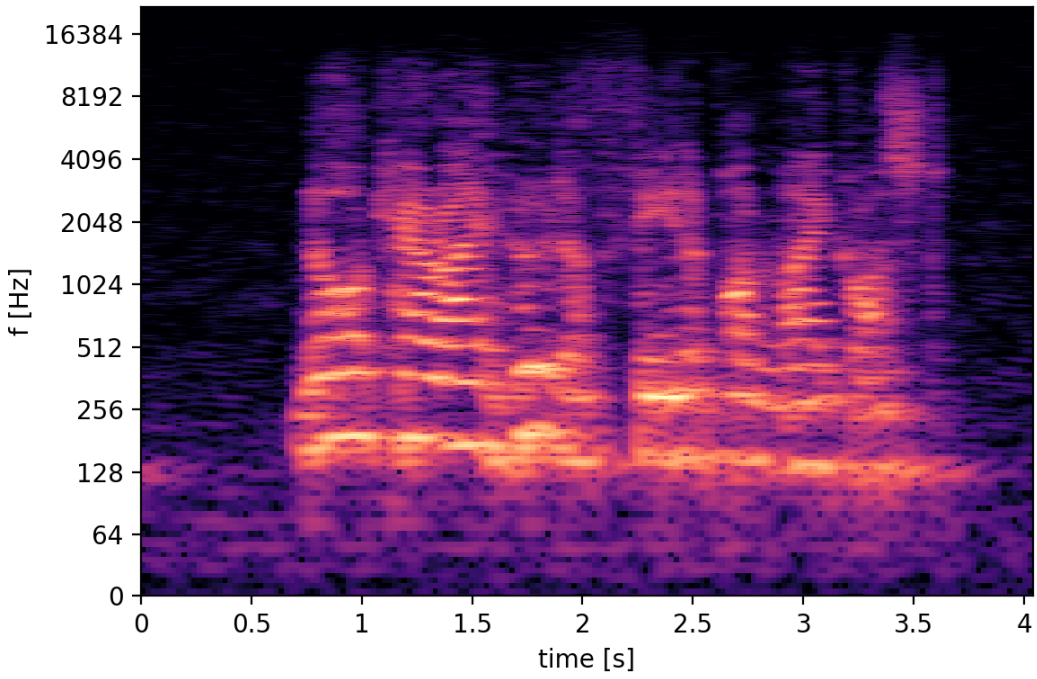

Figure 22 – Stage 2 with Rollers pitch-discretization.

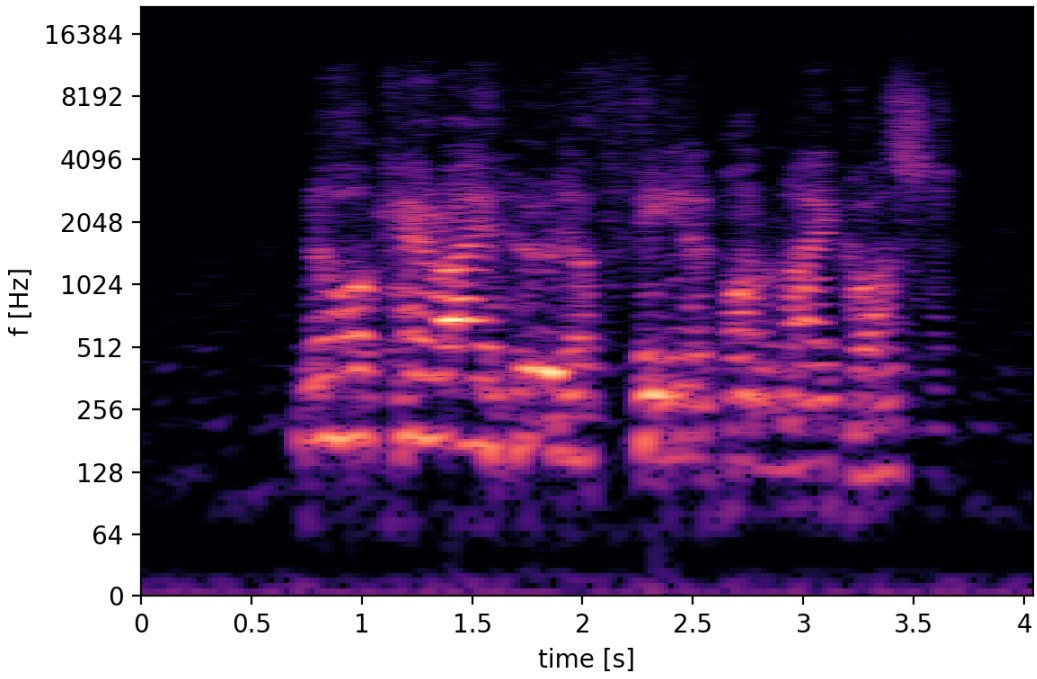

Figure 23 – Stage 4 with granular pitch-discretization.

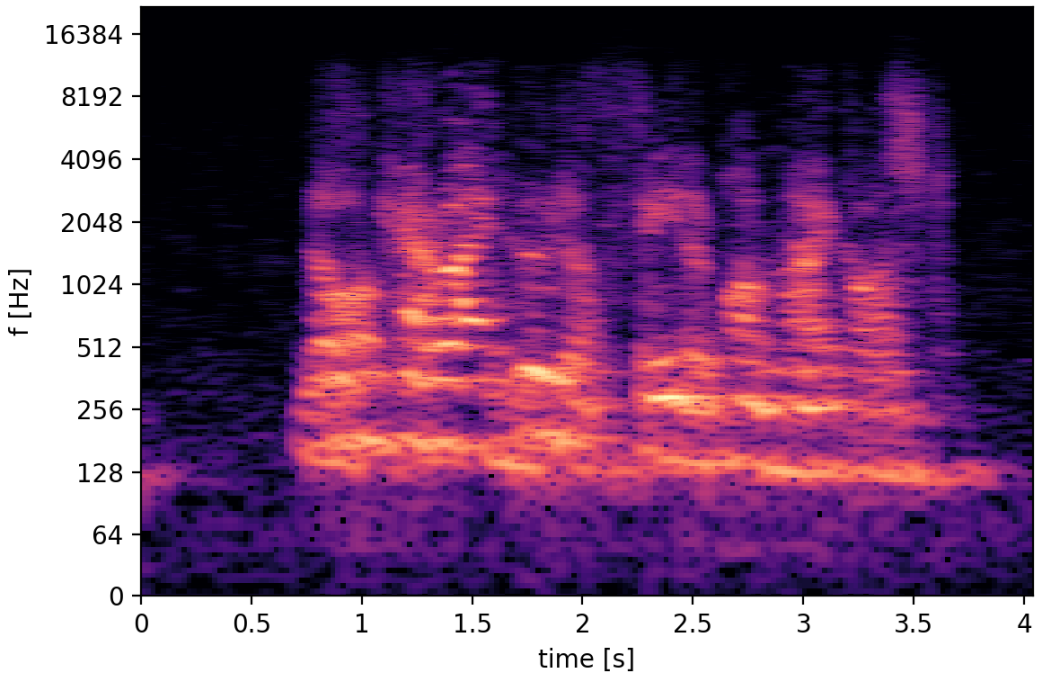

Figure 24 – Stage 4 with Rollers pitch-discretization.

5 Conclusion

Starting from the need of a real-time pitch-discretization effect using current algorithms to conduct an artistic experiment, lists of available pitch-shifting and pitch-tracking algorithms have been compiled and categorized. Furthermore comparative studies and current techniques have been presented. The different artifacts produced by pitch-shifting and possible improvements have been shown.

The general concept of pitch-discretization has been outlined and existing software has been mentioned. After proposing different pitch-discretization algorithm designs, a design for a real-time artistic purpose has been chosen for the actual implementation. Two versions have been implemented. Both of these versions used the *YIN* algorithm for pitch-tracking. One used pitch-shifting via *delay-line modulation* or *granular* pitch-shifting. The second effect used the *Rollers* algorithm for pitch-shifting. These three time-domain algorithms have been explained in detail and the resulting two effects implementations have been evaluated by processing a male speech signal and discussing the processed signal.

An artistic experiment using these two algorithms was conducted and the results have been analyzed. The Rollers algorithm produced stronger artifacts than the granular algorithm which manifested in strong resonances in the low end. The granular pitch-discretization effect produced a more evenly spread of voices. From one voice as input signal, every stage of the artistic experiment doubled the number of voices in the output signal. The pitch-discretization stages for monophonic speech signals jumped between these voices, since the algorithms dealt with a now polyphonic signal.

The artistic experiment resulted in different outcomes depending on the number of stages used. Original and pitch-discretized signals could be distinguished after one to four stages. From four to eight stages the resulting effect was similar to a chorus effect. Using more than ten stages resulted in an increasingly noise-like output signal, since the voices spread across the frequency spectrum.

For pitch-shifting, the Rollers algorithm was much more computationally expensive than the granular algorithm. This was due to the large filter bank that was not critically sampled, but used the original sample rate of the input signal. Downsampling of the subband signals was not an option, since this resulted in aliasing because the subband signals get frequency-shifted. In future implementations, a more efficient filter bank or a less expensive implementation may be used.

Although the developed effects were designed for monophonic speech signals, they can be used for any monophonic signal. The YIN pitch-tracking algorithm works well on a wide range of signals and the employed pitch-shifting algorithms offer reasonable sound quality and produce characteristic artifacts depending on the input signal and on their settings. In future artistic experiments, these produced artifacts could be exploited to generate interesting results. For instance, using the Rollers algorithm with few subbands may result in interesting frequency-shifting effects. Further artistic experiments are possible with this implementations and different effects could be achieved by using delays and feedback paths.

References

[Adr] F. Adriaensen, "Zita-at1," https://kokkinizita.linuxaudio.org/linuxaudio/zita-at1-doc/quickguide.html, accessed: 2020-09-30.

[Ant] "Antares auto-tune," https://www.antarestech.com/product/auto-tune-pro/, Antares, accessed: 2020-09-30.

[AR19] L. Ardaillon and A. Roebel, "Fully-convolutional network for pitch estimation of speech signals," in *Insterspeech 2019*, Graz, Austria, Sep 2019. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02439798

[Aub] "Graillon," https://www.auburnsounds.com/products/Graillon.html, Auburn Sounds, accessed: 2020-09-30.

[Bar] T. Baran, "Autotalent," http://tombaran.info/autotalent.html, accessed: 2020-09-30.

[BDd+13] O. Babacan, T. Drugman, N. d'Alessandro, N. Henrich Bernardoni, and T. Dutoit, "A comparative study of pitch extraction algorithms on a large variety of singing sounds," in *ICASSP 2013*, ser. 38th International Conference on Acoustics, Speech, and Signal Processing, 05 2013, pp. 7815–7819.

[Boe93] P. Boersma, "Accurate short-term analysisof the fundamental frequencyand the harmonics-to-noise ratio of a sampled sound," in *IFA Proceedings 17*, 1993, pp. 97–110.

[Cel] "Celemony melodyne," https://www.celemony.com/en/melodyne/what-is-melodyne, Celemony, accessed: 2020-09-30.

[CH07] A. Camacho and J. Harris, "A sawtooth waveform inspired pitch estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 124, pp. 1638–52, 2007.

[CK02] A. Cheveigné and H. Kawahara, "Yin, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, pp. 1917–30, 2002.

[DA11] T. Drugman and A. Alwan, "Joint robust voicing detection and pitch estimation based on residual harmonics," in *INTERSPEECH 2011*, ser. 12th Annual Conference of the International Speech Communication Association, vol. 12, 2011.

[DM16] J. Driedger and M. Müller, "A review of time-scale modification of music signals," *Applied Sciences*, vol. 6, no. 2, p. 57, Feb 2016. [Online]. Available: http://dx.doi.org/10.3390/app6020057

[dSiC20] O. das, J. o. Smith iii, and C. Chafe, "improved real-time monophonic pitch tracking with the extended complex kalman filter," *journal of the audio engineering society*, vol. 68, no. 1/2, pp. 78–86, january 2020.

[g20] "Kerovee," https://www.g200kg.com/, g200kg Music and Software, accessed: 2020-09-30.

[Ger03] D. Gerhard, *Pitch Extraction and Fundamental Frequency: History and Current Techniques*. Department of Computer Science, University of Regina, Regina, Canada, 11 2003.

[GFR+20] B. Gfeller, C. Frank, D. Roblek, M. Sharifi, M. Tagliasacchi, and M. Velimirovic, "Spice: Self-supervised pitch estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, p. 1118–1128, 2020. [Online]. Available: http://dx.doi.org/10.1109/TASLP.2020.2982285

[JH10] N. Juillerat and B. Hirsbrunner, "Low latency audio pitch shifting in the frequency domain," in *2010 International Conference on Audio, Language and Image Processing*, 2010, pp. 16–24.

[JSA08] N. Juillerat, S. Schubiger, and S. Arisona, "Low latency audio pitch shifting in the time domain," in *2008 International Conference on Audio, Language and Image Processing*, 08 2008, pp. 29–35.

[KEF01] H. Kawahara, J. Estill, and O. Fujimura, "Aperiodicity extraction and control using mixed mode excitation and group delay manipulation for a high quality speech analysis, modification and synthesis system straight," *Second International Workshop on Models and Analysis of Vocal Emissions for Biomedical Applications*, 09 2001. [Online]. Available: https://www.isca-speech.org/archive/maveba_2001/mv01_059.html

[KSLB18] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 161–165.

[LJ17] M. Lenarczyk and A. Janicki, "Voice conversion with pitch alteration using phase vocoder," in *2017 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, 2017, pp. 73–77.

[LK04] N. P. Lago and F. Kon, "The quest for low latency," in *PROCEEDINGS OF THE INTERNATIONAL COMPUTER MUSIC CONFERENCE (ICMC2004*, 2004, pp. 33–36.

[MD14] M. Mauch and S. Dixon, "Pyin: A fundamental frequency estimator using probabilistic threshold distributions," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 659–663.

[MW05] P. Mcleod and G. Wyvill, "A smarter way to find pitch," in *In Proceedings of the International Computer Music Conference (ICMC'05*, 2005, pp. 138–141.

[NHY11] M. Nagahara, K. Hanibuchi, and Y. Yamamoto, "Pitch shifting by $H\infty$-optimal variable fractional delay filters," in *Proceedings of the 18th World Congress*. The International Federation of Automatic Control, 2011.

[PDLCMS01] P. P. De La Cuadra, A. Master, and C. Sapp, "Efficient pitch detection techniques for interactive music," *International Computer Music Conference (ICMC)*, 01 2001.

[PG79] M. Piszczalski and B. A. Galler, "Predicting musical pitch from component frequency ratios," *The Journal of the Acoustical Society of America*, vol. 66, no. 3, pp. 710–720, 1979. [Online]. Available: https://doi.org/10.1121/1.383221

[Pla21a] M. Planton, "Pitch-discretization experiment," https://www.youtube.com/watch?v=o4CzSDUZVtY, 05 2021, accessed: 2021-06-10.

[Pla21b] ——, "Reshift pitch-discretization," https://git.iem.at/s1061531/reshift, 06 2021, accessed: 2021-06-29.

[Puc07] M. Puckette, *The Theory and Technique of Electronic Music*. World Scientific Publishing Co. Pte. Ltd., 2007.

[RB19] A. Rai and B. D. Barkana, "Analysis of three pitch-shifting algorithms for different musical instruments," in *2019 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, 2019, pp. 1–6.

[Sal] J. Salwen, "Talentedhack," https://github.com/jeremysalwen/TalentedHack, accessed: 2020-09-30.

[Sha02] W. Shakespeare, *Henry V*. Cambridge University Press, 2002.

[Skl06] A. Sklar, "A wavelet-based pitch-shifting method," 2006.

[SKS13] C. Schörkhuber, A. Klapuri, and A. Sontacchi, "audio pitch shifting using the constant-q transform," *journal of the audio engineering society*, vol. 61, no. 7/8, pp. 562–572, july 2013.

[Smi21] J. O. Smith, *Physical Audio Signal Processing*, 2010th ed. http://ccrma.stanford.edu/~jos/pasp/, accessed 2021, online book.

[Smi17] ——, *Spectral Audio Signal Processing*. http://ccrma.stanford.edu/~jos/sasp/, accessed 2020-09-17, online book, 2011 edition.

[SNJ+19] L. Shi, J. K. Nielsen, J. R. Jensen, M. A. Little, and M. G. Christensen, "Robust bayesian pitch tracking based on the harmonic model," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 11, pp. 1737–1751, 2019.

[Stu21] Stuendle, "Hilbert transform," https://commons.wikimedia.org/wiki/File:Hilbert-transfer-function.svg, 05 2021, accessed: 2021-05-16, public domain.

[Tal95] D. Talkin, *Speech Coding and Synthesis*. Elsevier Science B.V., 1995.

[vdKZ10] A. von dem Knesebeck and U. Zölzer, "Comparison of pitch trackers for real-time guitar effects," in *DAFx-10*, ser. Proc. of the 13th Int. Conference on Digital Audio Effects (DAFx-10), 09 2010, pp. 266–269.

[vdKZZ10] A. von dem Knesebeck, P. Ziraksaz, and U. Zölzer, "high quality time-domain pitch shifting using psola and transient preservation," *journal of the audio engineering society*, vol. 129, november 2010.

[Yea] G. Yeadon, "Gsnap," https://www.gvst.co.uk/gsnap.htm, accessed: 2020-09-30.

[Zö11] U. Zölzer, *DAFX: Digital Audio Effects, Second Edition*, 2nd ed. John Wiley & Sons, Ltd, 2011.