Friedrich-Alexander-Universität Erlangen-Nürnberg

Bachelor Thesis

Real-Time Beat Tracking for Creative Music Production

submitted by

Ole Frederik Müermann

submitted

July 30, 2024

Supervisor / Advisor

Peter Meier, M.Eng.
Prof. Dr. Meinard Müller

Reviewers

Prof. Dr. Meinard Müller

International Audio Laboratories Erlangen
A joint institution of the
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and
the Fraunhofer-Institut für Integrierte Schaltungen IIS.

Erklärung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Erlangen, July 30, 2024

Ole Frederik Müermann

Acknowledgements

I would like to express my gratitude to all people who supported me during the time of writing my thesis.

Especially I would like to thank my professor, Meinard Müller, my mentor, Peter Meier, and everyone working at AudioLabs who has greatly supported me. I am deeply grateful for the opportunity to write my thesis at AudioLabs in Erlangen. With their continuous close mentoring and support for my ideas, this thesis has been made possible!

Beyond the academic support, I would like to thank the creative individuals involved in my projects, including fellow DJs, music producer friends, and all music lovers. Thank you for your honest feedback! I want to express my gratitude to Michel Iseneld at Hypnus Records for providing access to the full “Slink Devices” collection, which are distributed in an official partnership with Ableton Live. A shoutout goes to Georg Kallasch at BR1 Radio for the inspirational song recommendations and the valuable insights he provided.

Last but not least, I want to thank my parents, family, siblings and friends for their support during the last months.

Abstract

Music production and mixing involve the use of numerous audio effects, with parameters that need to be controlled. While manual control is possible, automation greatly enhances efficiency. This thesis presents a collection of creative digital audio effects and audio examples for real-time beat tracking. A new real-time procedure based on the predominant local pulse (PLP) method is introduced, as described in [13] and [15]. This beat tracker can extract two types of control signals: First, a low-frequency oscillation (LFO), and second, a confidence measure for beat stability. LFOs are used in audio signal processing where an oscillating signal with a frequency below the audible range (typically below 20 Hz) modulates various audio parameters, such as amplitude or filter frequency. The main focus of this thesis lies on a library of case studies on creative audio effects. Selected effects are presented using a demo track produced in the digital audio workstation (DAW) Ableton Live. The pre-generated control curves from the beat tracker are utilized to automate effect parameters synchronized with the beat of the music. Subsequently, the audio examples are demonstrated using Jupyter Notebooks.

This study on creative music production aims to serve as a source of inspiration for music producers and sound engineers in potential applications and the further development of real-time beat tracking systems.

Zusammenfassung

Musikproduktion und -mixing umfassen die Verwendung zahlreicher Audioeffekte, deren Parameter gesteuert werden müssen. Durch die sich ergebenden grenzenlosen Möglichkeiten bei manueller Steuerung kann dabei die Effizienz durch Automatisierung erheblich erhöht werden. Diese Arbeit präsentiert eine Sammlung kreativer digitaler Audioeffekte und Hörbeispiele für das Echtzeit-Beat-Tracking. Ein neues Echtzeitverfahren basierend auf dem predominant local pulse (PLP)-Verfahren wird eingeführt, wie in [13] und [15] beschrieben. Dieser Beat-Tracker kann zwei Arten von Kontrollsignalen extrahieren: Ein LFO (Low-Frequency Oscillation) basiertes und zweitens Konfidenzkurven für die Beatstabilität. LFOs werden in der Audiosignalverarbeitung verwendet, wobei ein oszillierendes Signal mit einer Frequenz unterhalb des hörbaren Bereichs (typischerweise unter 20 Hz) verschiedene Audioparameter wie Amplitude oder Filterfrequenz moduliert. Der Schwerpunkt der Arbeit liegt auf einer Bibliothek von Fallstudien zu kreativen Audioeffekten. Anhand eines in der digitalen Audio-Workstation (DAW) Ableton Live produzierten Demo-Tracks werden ausgewählte Effekte aufgeführt. Dabei werden die vorgenerierten Kontrollkurven des Beat-Trackers verwendet, um Effekt-Parameter synchron mit dem Beat der Musik zu automatisieren. Anschließend werden die Audiobeispiele mithilfe von Jupyter Notebooks demonstriert.

Diese Studie zur kreativen Musikproduktion soll als Inspirationsquelle für Musikproduzenten und Toningenieure in potenziellen Anwendungen und der Weiterentwicklung von Echtzeit-Beat-Tracking-Systemen dienen.

Contents

Chapter 1

Introduction

We all find ourselves singing along with harmonies or tapping and dancing to music; it is in human nature. In music, a piece often consists of different instruments playing simultaneously. While it may be easy and natural for humans to tap along to a song, defining extraction features in algorithms that result in the expected tempo information is rather complex. The task of tempo and beat tracking focuses on retrieving onset information of relevant components in the music, contributing to the beat and rhythmic elements of a track.

The generation of reliable beatgrid information is automated in real-world software like Rekordbox [5] or MixedInKey [3] and is used for the annotation of large electronic music libraries. However, these programs require offline access to the whole audio track for analysis. This is where the contents of this thesis fit in: with the concept of a reliable, zero-latency real-time beat tracking system as described in [13], [14], and [15], new creative possibilities arise. The data not only includes past and future predicted pulse periodicity information in the form of low-frequency oscillation (LFO) shapes but also a beat stability measure extracted from an input audio signal. LFOs are utilized in audio signal processing to modulate various audio parameters, such as amplitude or filter frequency, by generating oscillating signals with frequencies below the audible range (typically under 20 Hz). In the context of this thesis, LFOs play a crucial role in shaping control signals that synchronize audio effects with the beat of the music.

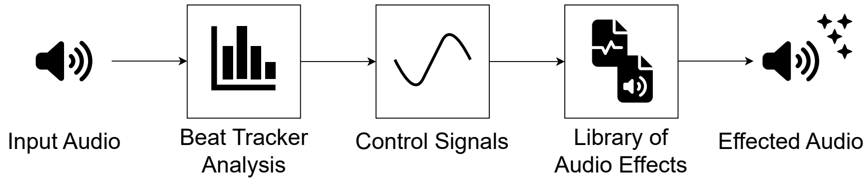


Figure 1.1. The input audio signal undergoes beat tracking analysis to extract beat and tempo information. Control signals are generated, which automate a selected audio effect inside the DAW. The final output is the audio signal enhanced with effects synchronized with the detected beat.

With the ability to extract beat and tempo data from audio signals, new automation possibilities in an adaptive music production environment are explored. To demonstrate these possibilities, pre-generated control signals are used. The main scope and objective of this thesis are to create a creative digital audio effects library with sample demonstrations, aimed at the application field of real-time beat tracking. This library is designed with the creative music producer and mixing engineer in mind.

This thesis is organized as follows: first, the basic technical foundations for extracting beat and pulse information from an audio signal as well as relevant audio representations based on the Fourier Transform will be described in Chapter 2. This chapter builds upon the sources [15] and [13]. Resting upon this, in the following the foundations for tempo analysis and an offline beat tracking approach are outlined. After that, the concept for an online beat tracker is presented. The main parts of the thesis are as follows. Feasible effects and signal modulations based on the retrieved LFO and confidence outputs of the beat tracker are outlined in Chapter 3. Within this, integration possibilities into the creative music producers and mixing engineers workflow are depicted. Figure 1.1 illustrates the overall process of beat tracking and effect application. The thesis finishes with Chapter 4 on creative music production, including pre-produced audio examples from one demo track and visualized scenarios accompanied by a Python Jupyter Notebook for demonstration purposes outside of a classical digital audio workstation. Chapter 5 summarizes the key findings of this thesis, discusses the implications of the research, and suggests directions for future work. Additionally, the Appendix provides supplementary materials, including code snippets and further case studies.

During the writing of this thesis I relied on multiple tools for assisted correction and spell checking, which I want to acknowledge here. The ChatGPT AI was prompted to correct selected text passages with minimal spelling adjustments, and a check through Grammarly was applied to sections.

Chapter 2

Theoretic Foundations

In the field of Music Information Retrieval (MIR), many foundational tasks are equally important across different special fields. This thesis focuses on beat tracking, which relies on extracting information from a musical piece contributing to tempo and beat. This chapter on theoretic foundations covers audio feature representations and concepts required to analyze an audio signal with a focus on tempo and rhythmical aspects, following the descriptions from [16] and [17] along with example figures. The chapter concludes with an offline beat tracking approach, which relies on the predominant local pulse (PLP) method and is then extended to a real-time beat tracking system. Different control signals representing pulse oscillation and stability measures are extracted based on the concepts presented in [13].

2.1 Audio Representation

An audio signal is a digital encoding of sound and depicts how air molecules vibrate in different waves of pressure over time. It differs from sheet music and MIDI-representations, that are considered a well known standard notation [12]. In these two forms specific temporal and harmonic aspects of instruments playing in polyphonic music are noted. From an audio signal this can not be directly determined.

In Figure 2.1, the waveform (a), spectrogram (b), and novelty function (c) of a 4-bar drum pattern are illustrated. The audio amplitude in 2.1a varies over time showing the fluctuations in the audio signal's volume. Peaks and troughs indicate louder and quieter sections. The waveform shows periodic patterns, suggesting rhythmic or repeating elements. To analyze this observation, one can define technical terms to identify data from the signal.

Relevant for extracting information from a musical piece contributing to tempo and beat is the note onset. The onset is defined as the beginning of the transient and is a noise-like sound

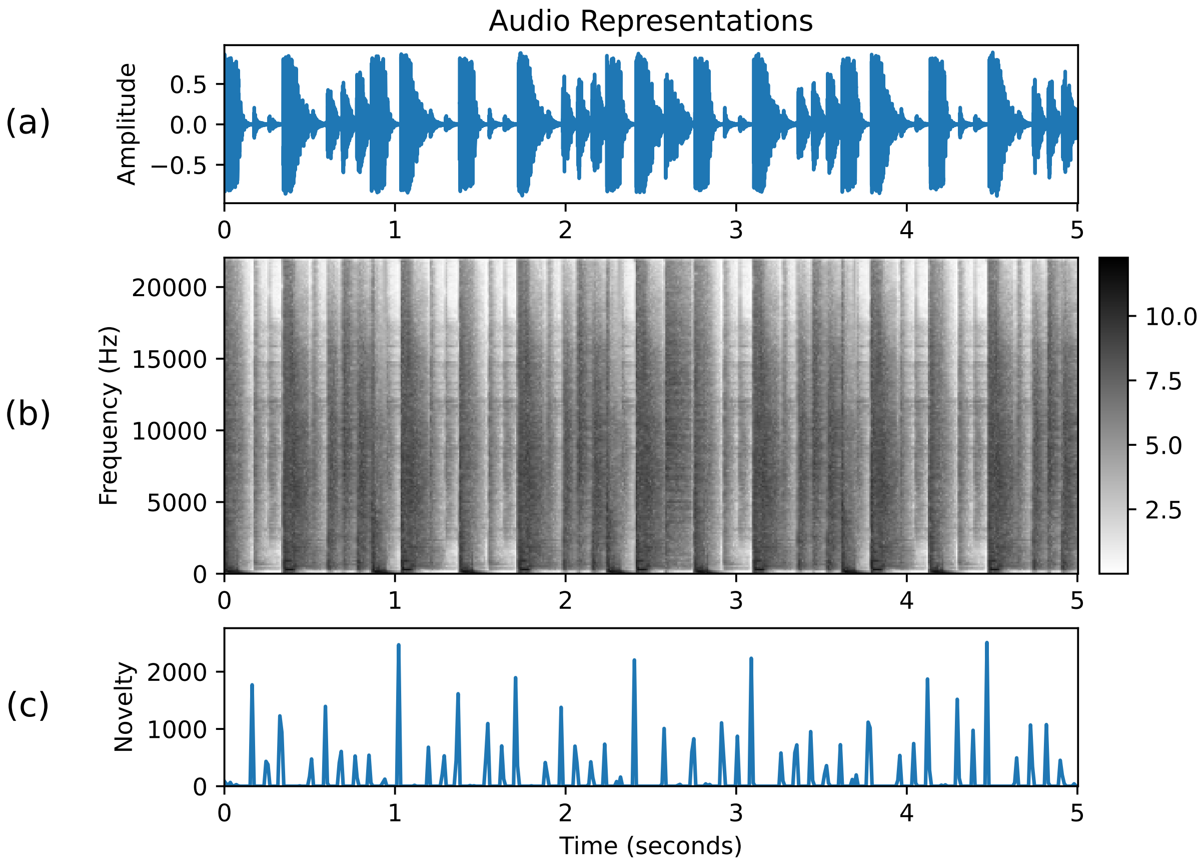


Figure 2.1. Waveform, spectrogram, and novelty function of a 4-bar drum pattern.

component of short duration and high amplitude, typically occurring at the beginning of a musical tone [16]. Most drums have a clearly distinguishable onset and transient, while in a violin solo, you may not detect clear transients. The difference between percussive instruments (contributing to rhythmic parts) and harmonic instruments (contributing to melodies in a piece) can be illustrated within another representation, the spectrogram, which yields a time–frequency representation of an audio signal [16].

The spectrogram pictured in 2.1b is based on the mathematical concept of the short-time Fourier transform (STFT), which converts a signal from the time domain to the frequency domain. Contributing to the beat is the non-harmonical drum onset information, visible as darker vertical bands of higher intensity at regular intervals in the STFT spectrogram, suggesting a repeating rhythmic pattern with prominent frequencies.

To visualize possible note onset candidates, the notion of a novelty function measuring spectral changes over time is introduced as shown in 2.1c. Peaks in the novelty curve indicate points where significant changes or new elements occur in the audio, and correlate with the periodic positions of the vertical lines in the spectrogram [14].

2.2 Tempo Analysis

With the goal to achieve consistent beat periodicity and tempo analysis over the duration of a musical piece, we have to deal with irregularities caused by the composers intentions, the origin of the genre (instrumental vs. electronic music), and the performers artistic interpretation. Intentional and unintentional disruptions of a strict beat pattern are widely spread. A well-known example is Queens "Bohemian Rhapsody", where the song begins with a slow tempo and features significant tempo changes over time.

To visualize global tempo properties of a musical piece, Fourier analysis is applied on the novelty function to detect periodic patterns, resulting in a tempogram as pictured in Figure 2.2 [16]. This notion relies on two assumptions: that beat positions go along with note onsets, and that beat positions are periodically spaced [13]. The tempogram shows tempi of high and low correlation, measured in beats per minute (BPM) over time (in seconds), similar to the spectrogram representation of relevant frequencies over time. Intuitively, it depicts to which extent windowed sinusoids of different frequencies overlap with the periodic patterns of the novelty function, illustrating the tempo correlation at time position t. For example, a novelty section with periodically distributed peaks with period $T = 3/4$ sec $= 0.75$ sec corresponds to a rate of $\omega = 1/T = 4/3$ Hz $= 1.\overline{3}$ Hz or a tempo of 80 BPM [16].

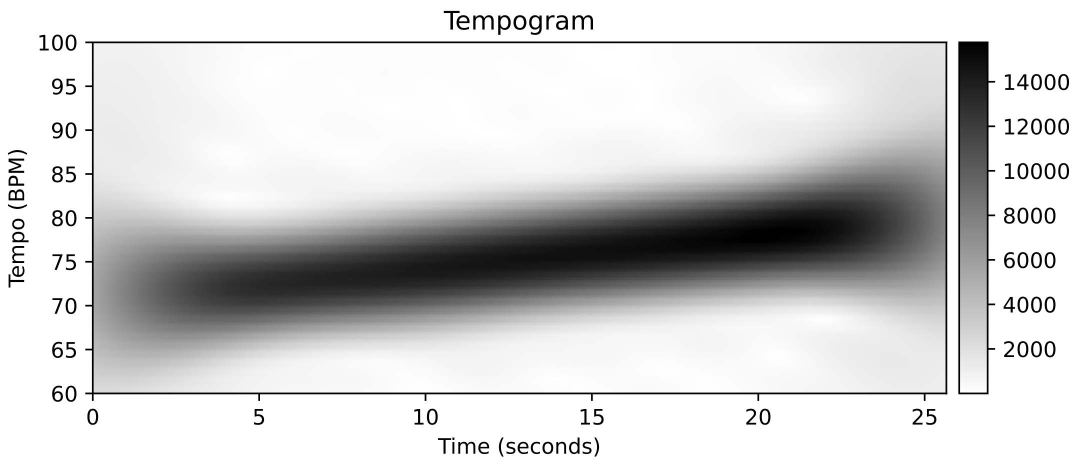


Figure 2.2. Tempogram displaying regions of high and low intensity, with darker areas indicating more dominant tempi at specific time positions. The input audio is a click track with linear tempo increase starting from 70 BPM and rising to 80 BPM.

The decision on limiting the BPM range for tempo estimation in a song is recommended to be made individually for music of different genres. A more specific selection of suggested parameters is listed in Table 2.1. Using the broad tempo range from 70 BPM - 180 BPM to analyze the demo pictured in the tempogram 2.3, the result can vary. While estimations in software like Rekordbox [5] widely used in the music industry are precise most of the time, DJs and Music Producers have to be aware of the mentioned fluctuating analysis results: a tune at 160 BPM is technically suitable to play along with one at 80 BPM, because they are multiples. The difference between half or double the tempo is called a tempo octave, which can be observed in example Figure 2.3 where two prominent dark bands are visible.

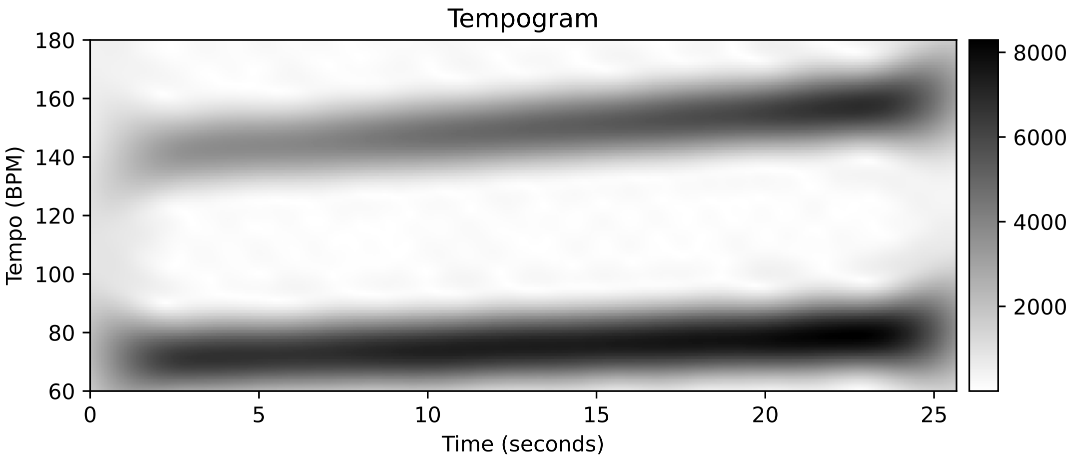


Figure 2.3. Tempogram displaying regions of high and low intensity, with darker areas indicating more dominant tempi at specific time positions. The input audio is a click track with linear tempo increase starting from 70 BPM and rising to 80 BPM. Visible is the tempo octave ranging from 140 BPM to 160 BPM.

This short demo pictured in Figure 2.3 was composed in the range of 70 BPM - 80 BPM displayed in the lower band, but the tempo octave between 140 BPM and 160 BPM is also clearly detected. In musical terms this can be explained with the tatum (temporal atom), tactus (quarter note) and measure (one bar) level that often are multiples of each other. It is advisable to set the tempo range to 160 BPM - 180 BPM as indicated in Table 2.1 when analyzing Drum and Bass tracks. This specific limitation helps in avoiding tempo octave confusion and ensures a more accurate tempo detection within the genre's common BPM range.

Tempo Range (BPM)	Genre
88 - 175	Hip-hop, Drum and Bass
70 - 180	Broad range (general use, less precise)
100 - 140	House, Techno
120 - 170	House, Drum and Bass
160 - 180	Drum and Bass
128 - 255	Double time for various genres

Table 2.1. Table with a selection of suggested BPM settings for tempo analysis inspired by the DJ software Rekordbox [5]. Note that the different parameter pairs in- or exclude tempo octaves. Some settings are intended for analysis of a song library with multiple genres (resulting in songs being categorized in the wrong tempo octave) and other settings are more genre-specific and well-suited with better results.

Besides global tempo information in the tempogram, one can also reveal local periodic pulse patterns that can be used for beat tracking applications. This can be done by applying postprocessing on the novelty function. The process for this is pictured in Figure 2.4. For the input audio shown in 2.4a, its corresponding spectrogram (pictured in 2.4b) and novelty function (shown in 2.4c) are calculated. The novelty function is compared with locally windowed sinusoidal kernels (see Figure 2.4e). For each time position, the optimal match of the sinusoidal kernels with the local tempo structure of the signal within a specified BPM range is calculated. These optimal time positions are visualized as colored dots in the tempogram (2.4d). A global predominant local pulse (PLP) function (see Figure 2.4f) is retrieved by overlap-adding all optimal kernels and applying normalization, so that the values of the PLP function fall within the range of [0:1] [16].

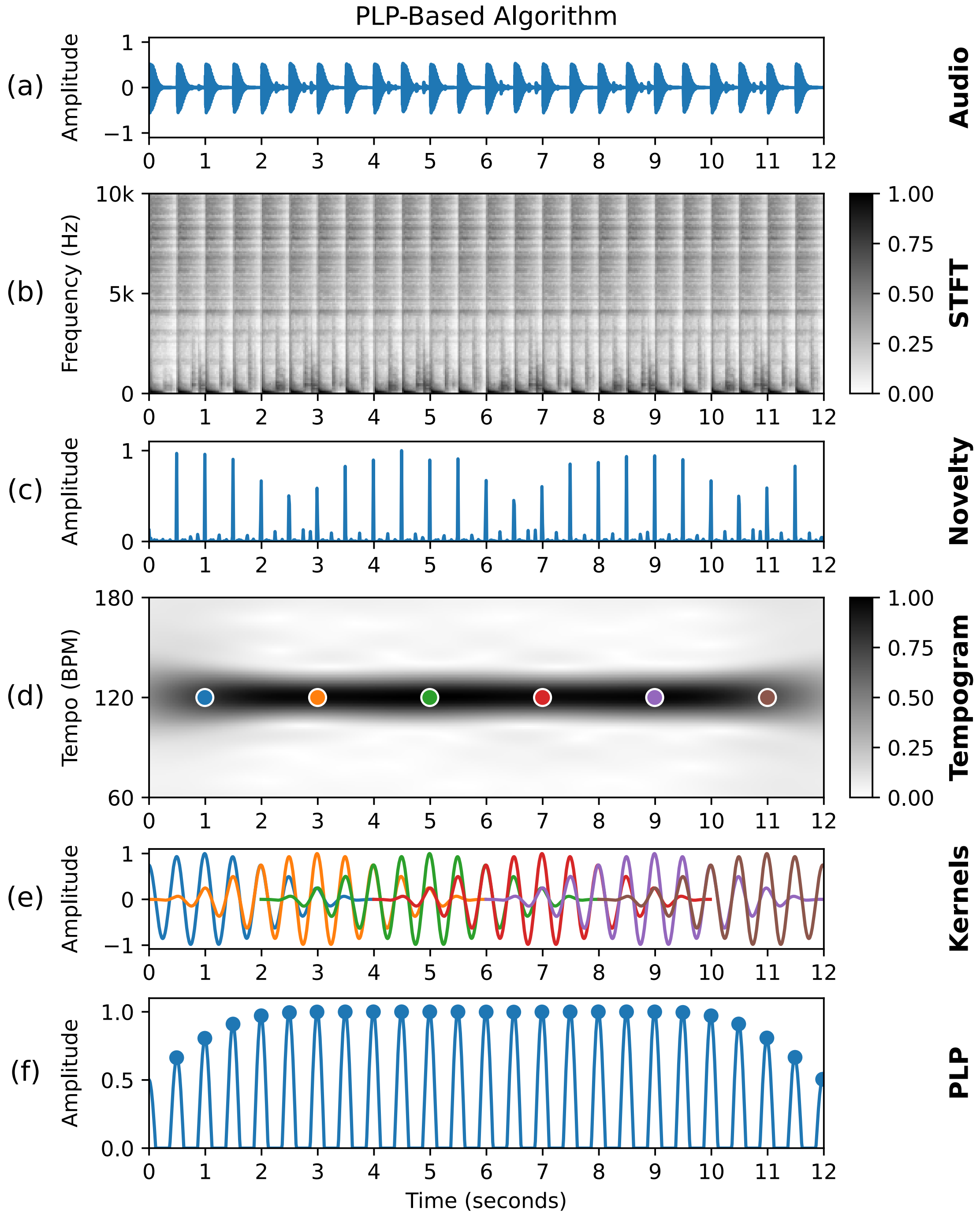


Figure 2.4. Step-by-step offline PLP pipeline, based on the figure found in [13]: (a) Waveform. (b) Spectrogram. (c) Novelty. (d) Tempogram. (e) Kernels. (f) PLP. This offline concept was originally introduced in [9].

2.3 Real-Time Beat Tracking

Live performances and shows create a need for real-time beat tracking applications to enable new creative possibilities. Existing methods for beat tracking typically operate offline and thus require access to the entire music track for processing [13]. In contrast, a real-time beat tracker has access only to the audio data at the current time position and past data. Converting from an offline procedure to a reliable online procedure involves addressing challenges related to latency issues and controllability. In the following, the concept for a zero-latency beat tracker is presented. Additionally, the methods for extracting four different control signals divided into pulse oscillation and stability measures are discussed.

With Figure 2.7, the pipeline of a real-time PLP procedure is described, closely following the sources found in [13] and [15]. First, an audio signal 2.7a is transformed into its corresponding novelty function 2.7c via the Fourier spectrogram 2.7b to detect local periodic patterns. Windowed kernels centered around different time positions can be split into two halves: the left side shows data from the past and present, the right side shows predicted pulse information in the upcoming audio signal. By overlap-adding kernels (see 2.7e), future beat positions can be estimated, to form a PLP buffer windowed around the current time position $t = 0$ (see Figure 2.7f).

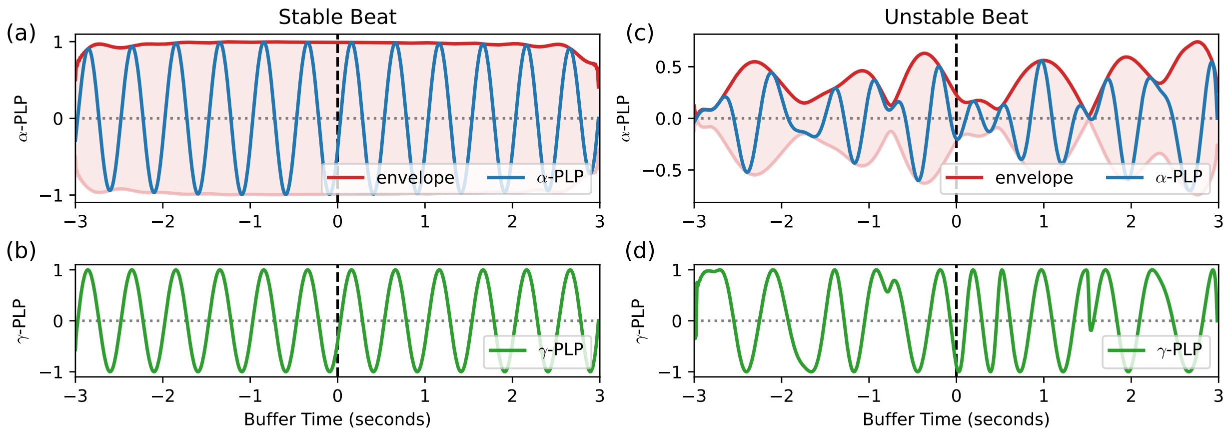


Figure 2.5. Stable and unstable PLP curve as illustrated in [15] and their envelopes as a result of constructive and destructive interference in comparison.

We now look at Figure 2.5. When the frequency of neighboring kernels is similar (indicating stable tempo of the audio signal), the sinusoids interfere constructively, leading to high PLP amplitude values, as shown in Figure 2.5a. Contrary, during periods of unstable tempo, the sinusoids interfere destructively and cancel each other out, as pictured in Figure 2.5c. These α-PLPs can be normalized to form a γ-PLP function that oscillate between values of -1 and 1. For a stable beat, the constant and high envelope can be observed in 2.5a. Conversely, unstable beats create a fluctuating envelope, as highlighted in 2.5d in red. The PLPs will now be further used to create a measure for pulse oscillation, and the envelopes will be used to create a confidence measure for beat stability.

We are looking at Figure 2.6 to explain four control signals representing pulse oscillation and beat stability data as follows. Shown on the left side are the audio input signal and the control curves. Visualized on the right side are the tempogram from the audio, as well as the PLP buffers with the central buffer position n_0 corresponding to the real time marked by the black cursor on the left. Beat stability measures named the β- and γ-confidence are extracted for the input audio 2.6a. The β-confidence control signal is pictured in 2.6b. The confidence at position n_0 is defined as the value of the envelope at the central buffer position. The γ-confidence shares similarities with the β-confidence, but it has a lower amplitude. The α-LFO is pictured in 2.6d. It includes a pulse oscillation as well as a beat stability measure. Its amplitude decreases when the beat is unstable, and conversely reaches high amplitude values limited to -1 and 1 when the beat is stable. Shown in 2.6e is the γ-LFO. In comparison to the α-LFO, the values of the γ-LFO are normalized to uniformly oscillate between values of $[-1 : 1]$, without the influence of beat stability.

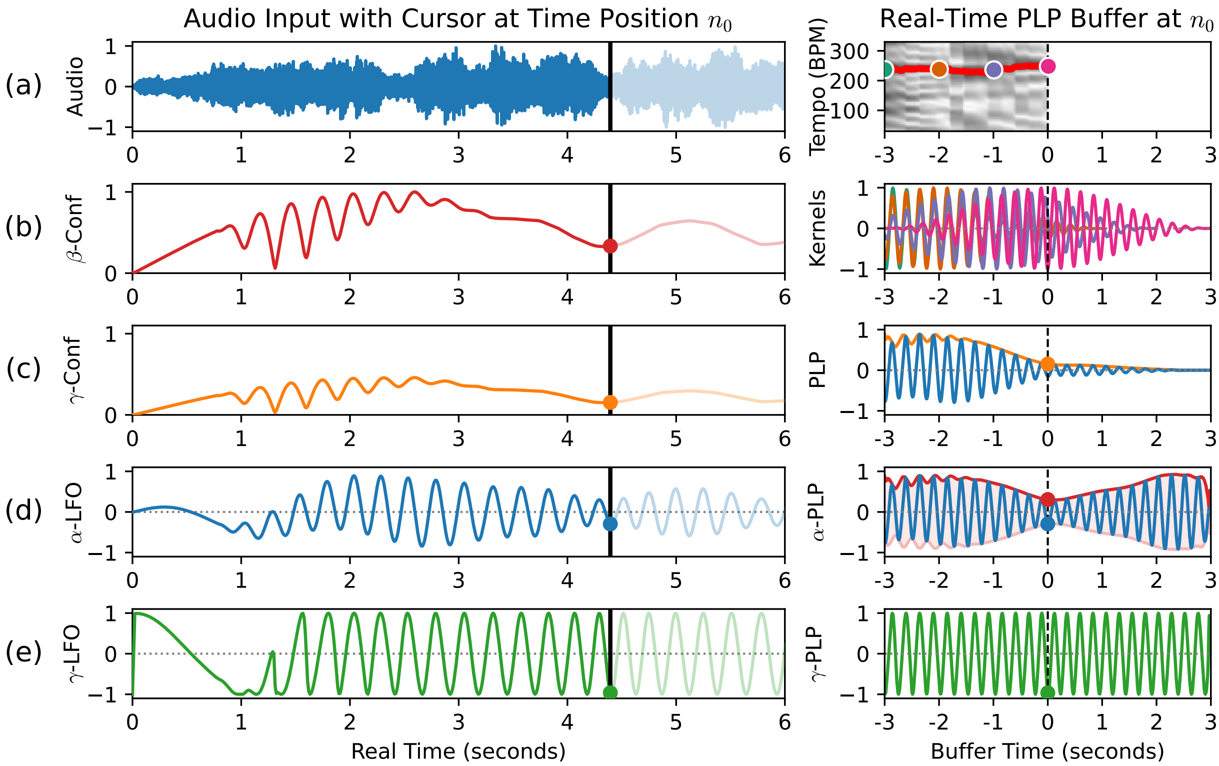


Figure 2.6. Real-Time control signals with cursor at current time position n_0 as described in [15]. (a) audio signal. (b) β-Confidence. (c) γ-Confidence. (d) α-LFO. (e) γ-LFO. On the right side, the PLP buffers at the time position marked by the black cursor are shown.

The two LFO control signals have their unique application scenarios. Depending on the controlled effect parameter, the beat confidence is included or excluded during effect control: if the parameter settings need to oscillate exactly between two fixed limits, the γ-LFO is suitable. If a confidence as well as a beat stability measure need to be applied at the same time, the α-LFO can be used. There are other scenarios where using confidence measures without included beat stability data is beneficial. The β-confidence is mostly used in this case, because of its higher amplitude values. Examples will be given in Chapter 3 and 4.

The software used to extract the control signals for later use in this thesis is run in Python. The generation and export to the `.wav` format is achieved with the script `wav2controlsignals.py` based on `beatcli.py` [13]. During generation, different parameter settings can be adjusted:

```
[--buffersize SAMPLES] [--tempo LOW HIGH] [--lookahead FRAMES] [--kernel SIZE]
```

The `buffersize` parameter determines the size of the buffer, as diplayed with the red rectangle in Figure 2.7f. With the `LOW` and `HIGH` tempo values, the BPM range of interest for analysis can be specified. For suggested settings see Table 2.1. Reliable for delay compensation and thus achieving zero-latency in a real-time scenario is the `lookahead` parameter. Lastly, the size of the used `kernels` is modyfiable to control the beat trackers responsiveness to tempo changes.

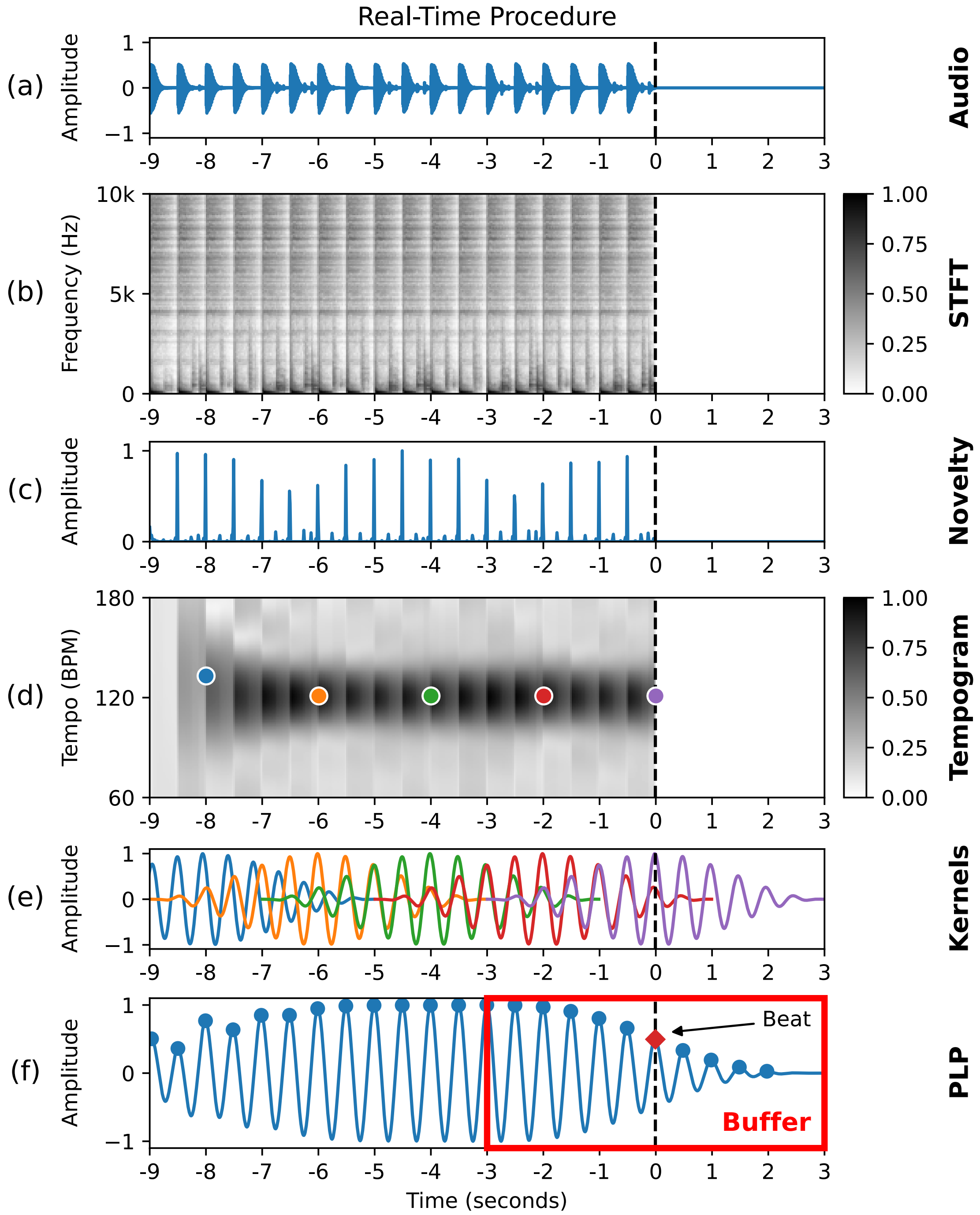


Figure 2.7. Step-by-step real-time procedure, based on the figure found in [13]: (a) Waveform. (b) Spectrogram. (c) Novelty. (d) Tempogram. (e) Kernels. (f) PLP.

Chapter 3

LFO and Confidence-Based Signal Modifications

Resting upon the LFO and confidence-based control signals that can be extracted by a real-time beat tracker, inside a DAW multiple options for mapping and automating parameters to effect the generated audio output can be explored. In this chapter, a collection of these modifications are outlined to later be demonstrated in case studies with produced audio examples using one or many of the techniques. The described effects can be easily adapted and applied to different elements within a production, offering creative opportunities for music producers and sound engineers. An introduction to commonly used terminology in this chapter and in the field of music production is provided in the glossary (see Appendix A).

3.1 Waveshaping

Automations in DAWs, based on signals changing over time, often come in a variety of non-sinusoidal periodic waveforms. One way to achieve this, is to shape the LFO control signal output of the real-time PLP system. For this, a customized version of the shaper plugin included in Ableton Live Suite (see screenshot in Figure 3.1) is used [1]. It enables the generation of modulation data for musical expression by creating a multi-breakpoint envelope. This envelope can be linked to various parameters within Ableton Live. The shaper plugin offers three main modes: `Loop`, which continuously repeats the envelope; `1-Shot`, which triggers the envelope once; and `Manual`, which allows direct control over the envelope. Additionally, the plugin supports adding randomness to the modulation and smoothing the curve of the envelope. The envelope speed can be synchronized with the tempo of the Live project [1].

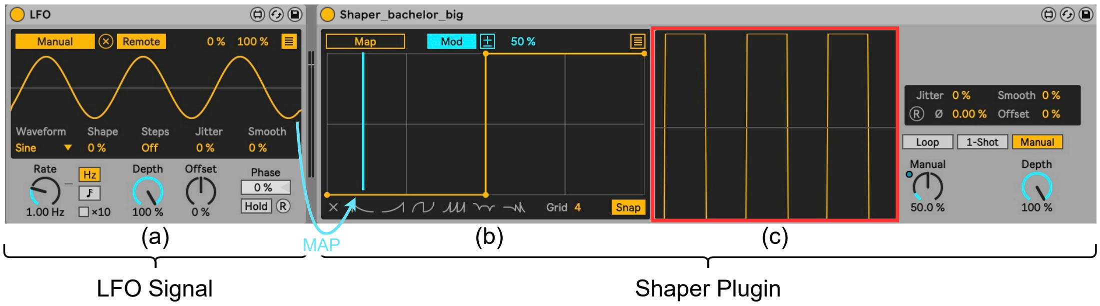


Figure 3.1. On the left (a) a simple sinusoidal LFO colored in yellow at the rate of 1 Hz is set. In the middle black window (b) inside the shaper plugin a square waveform shape is drawn. The amplitude of the LFO is mapped to the manual input of the customized shaper. Visualized by the blue line is the current region of the custom shape the LFO is mapped to: when the LFO amplitude is less than zero, the blue line in the drawn square waveform is in the region smaller than zero, resulting in the yellow square waveform in the rightmost red-highlighted window (c).

To better describe the connection between the application displayed in Figure 3.1 and the waveshaping process, a mathematical notation is introduced along Figure 3.2.

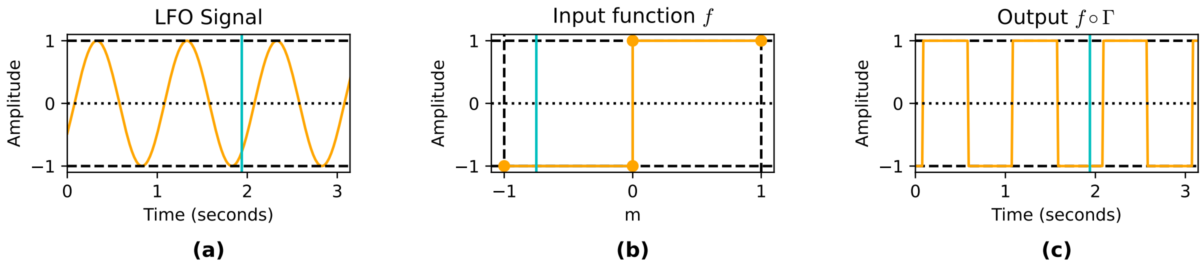


Figure 3.2. (a) Sinusoidal LFO at the rate of 1 Hz, with phase offset $\phi = -0.5$. (b) Step function f limited in both axis between [-1,1]. (c) Square waveform. The current time position is visualized by the blue line. When the LFO amplitude (a) is less than zero, the blue line in the step function (b) is in the region smaller than zero, resulting in the Square waveform (c).

The sinusoidal α-LFO can be described as a continuous function $\Gamma : \mathbb{R} \to [-1, 1]$ given by the equation

$$\Gamma(n) = \sin(\omega n + \phi) \tag{3.1}$$

where $n \in \mathbb{R}$, $\omega \in \mathbb{R}$ is the angular frequency, and $\phi \in \mathbb{R}$ is the phase offset, as illustrated in Figure 3.2a.

Illustrated in Figure 3.2b is the step function $f : [-1, 1] \to [-1, 1]$, given by the equation

$$f(m) = \begin{cases} 1 & m \geq 0, \\ -1 & \text{otherwise.} \end{cases} \tag{3.2}$$

for $-1 \leq m \leq 1,\ m \in \mathbb{R}$.

Using the continuous amplitude of Γ as an input value to the locally windowed step function f between $[-1, 1]$, the result is a continuous square waveform with values synced to the time positions n of the LFO sinus input:

$$(f \circ \Gamma)(n) = f(\Gamma(n)) \tag{3.3}$$

for $n \in \mathbb{R}$, as illustrated in Figure 3.1c.

Demonstrated with this simple sinusoidal to square waveform conversion, the principle can be applied to any arbitrary shape that is drawn in the input function window of the shaper plugin.

3.2 Rhythmic Manipulation - Arpeggiator

An arpeggiator can create harmonic structures from a single note by adding semitones over the defined input in custom patterns such as upwards, downwards, or custom sequences. It is a plugin included in most DAWs, as shown in the screenshot in Figure 3.3. The plugin can take one MIDI note and "arpeggiate" it at a given rate, which can be input as note values (1/4 notes, 1/8 notes) or milliseconds. Simplified, a new tone is played repeatedly at the pacing defined by the rate.

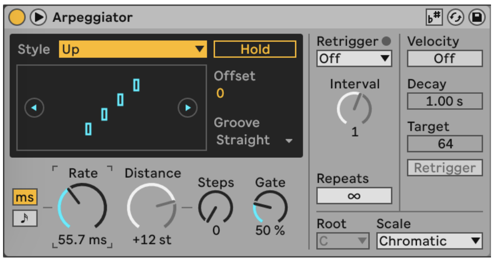


Figure 3.3. Example of an arpeggiator plugin inside Ableton Live. The arpeggiator is set to play a sequence of notes in an upward pattern at a rate of 55.7ms.

By using LFO or confidence signals to control the speed and pattern of the arpeggiator, complex rhythmic patterns can be created. The LFO control signals can be used to modulate the arpeggiator rate, creating dynamic changes in the rhythm over time. This is special because these rhythmic patterns are in sync with the beat of the music, due to the nature of the LFO signal being extracted from beat patterns. Additionally, confidence signals are suitable to control the intensity and speed of the arpeggiation based on the detected beat confidence.

An example of using an LFO signal mapping to control an arpeggiator is given in the case study in Chapter 4.2.1.

3.3 Volume Based Manipulations

The gain or volume of an audio signal can be adjusted in multiple ways to create effects in sync with the beat of the music. In the following, different techniques for gain based manipulations are discussed.

3.3.1 Sidechain

Sidechain compression is used to create a "pumping" effect by reducing the gain of a track when another track (usually a kick drum) is playing [10]. This technique is commonly used to control the instruments playing in the low frequencies in electronic dance music. The gain reduction is triggered by the sidechain input, typically the kick drum, creating a rhythmic pulsing effect that synchronizes with the beat. Typically this effect is created using a compressor, which is a plugin that first removes high amplitude peaks from the signal and then applies make-up gain to the whole signal for an overall increase in loudness. By reducing the dynamic range, the compressor helps in managing the loudness levels in an audio signal [20]. With LFO signals in sync to the kick drum and mapped to the gain control of an audio track, a similar effect can be created. An example of using an LFO signal mapping to create a sidechain pumping effect on the bass is given in the case study in Chapter 4.2.5.

3.3.2 Vocal Stutter or Tremolo

Vocal stutter effects are created by rapidly reducing and increasing the gain of the vocal track [10], [20]. This can be done by using an LFO to modulate the gain, creating rhythmic stutters that add a percussive element to the vocals. The LFO can control the rate and depth of the stutter effect, allowing for precise timing and intensity adjustments that sync with the beat of the music. This rhythmic variation of the volume could also be described as a tremolo, which is often used as a guitar effect. In the case study in Chapter 4.2.3, a demo of using an LFO signal mapping to create this described effect is provided.

3.3.3 Volume Mixing

Every audio track in a DAW has its own volume parameter, which controls the volume level of the track in the overall mix. A screenshot of the mixing section in Ableton Live 12 can be seen in Figure 3.4. Volume mixing involves adjusting the gain levels of different tracks to achieve a balanced and cohesive level during production. By using confidence signals, the mix can be dynamically adjusted based on the stability and intensity of the beat. During sections of high confidence elements like bass or lead instruments can be brought forward in the mix, while during low confidence sections ambient or background elements can be amplified.

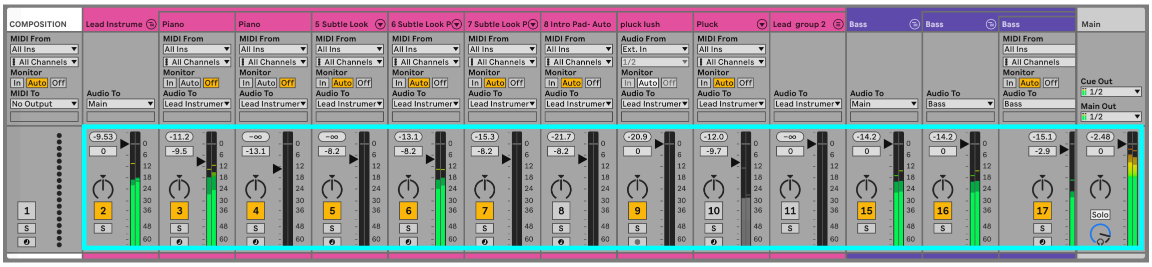


Figure 3.4. This screenshot displays the mixing section inside Ableton Live 12. Multiple tracks have their individual volume faders, as highlighted in the ligh blue box.

3.4 Wavetable Synthesis

In this section, the sound generation possibilities with synthesizers and manipulations based on confidence and LFO curves are discussed.

Figure 3.5. Screenshot of the Vital Synthesizer plugin. On the left border, user definable "Macro" dials are highlighted in blue. On the left side marked in red, three active oscillator curves are shown. In the top right green highlighted window, the ADSR envelope for oscillator one is pictured.

In wavetable synthesis, one of the powerful features is the ability to shape or manipulate the oscillator curve to create unique sounds [19]. The Vital Synthesizer plugin [7] offers various tools for adjusting the oscillator waveform, allowing for precise control over the sound's timbre and character.

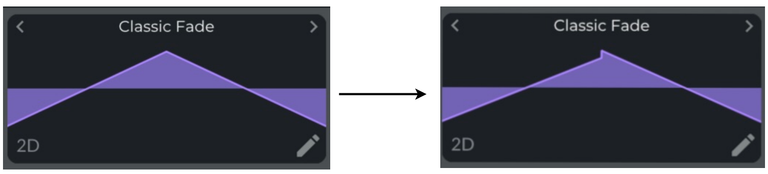


Figure 3.6. Example of an oscillator curve shaping in the Vital plugin. The left image shows the initial curve, and the right image shows the modified curve with additional points and shape adjustments. Although the adjustment is visually minor, the effects on the timbre are significant.

Figure 3.6 illustrates how the oscillator curve can be manipulated. The left image shows the initial "Classic Fade" waveform preset in the synthesizer, while the right image demonstrates the same waveform with minor adjustments. These modifications allow for the creation of custom waveforms that can be tailored to fit the desired sound. The Vital Synthesizer plugin also offers the ability to blend between multiple waveforms. This blending setting is a suitable parameter to be controlled by an LFO curve, allowing the generated timbre to change dynamically in sync with the beat. In Figure 3.5, the setting has been mapped to the user definable "Macro" dials on the left. An example of this blending technique is provided in Chapter 4.2.6.

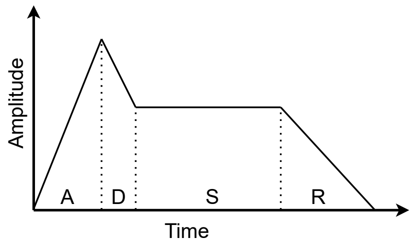


Figure 3.7. ADSR envelope, closely following the description and graphic found in [16]. Attack (A): This is the initial phase where the amplitude of the sound increases from zero to its maximum level. The duration of this phase is determined by the attack time. Decay (D): After the attack phase, the amplitude decreases to a sustain level. The decay time defines how long it takes for this decrease to occur. Sustain (S): This stage represents the duration during which the amplitude remains at a constant level after the decay phase. The sustain level is maintained as long as the note is held. Release (R): When the note is released, the amplitude decreases from the sustain level to zero. The release time determines how long this decrease takes.

There are many more parameters inside synthesizers that can be controlled by the LFO and confidence-based control signals. To give an outlook, another common setting is the the ADSR (Attack, Decay, Sustain, Release) envelope, as pictured in Figure 3.7. The ADSR envelope is also explained in the glossary Chapter A, like other relevant audio terms used. It can be manipulated using signals from the beat tracker. By adjusting the ADSR parameters based on the confidence level, various sound textures can be created. For example, a higher confidence level leads to an increased attack and release time, resulting in a more atmospheric sound. Conversely, a lower confidence level shortens these times, producing a tighter and more percussive sound. Besides manipulations based on confidence control signals, the ADSR envelope can also be modified with the LFO signals from the beat tracker to create rhythmic variations in the sound. The attack time can be affected from a low to a high corresponding LFO value to create a pulsing effect. Additionally, the release time can be modulated to create a gating effect. This manipulation of the ADSR envelope allows for the creation of complex and evolving synthesizer effects in sync with the beat.

3.5 Timbre and Frequency-Based Effects

3.5.1 Formant Shifting

Formant shifting involves altering the resonant frequencies of a sound to change its perceived vocal characteristics. Formants are the frequency bands that define the tonal quality of a sound, particularly in human speech and singing. They are responsible for the distinct vowel sounds and are created by the resonances of the vocal tract. In audio processing, formant shifting can be used to modify these resonant frequencies without changing the pitch of the sound, resulting in a change in timbre or color of the voice [20]. This can make a voice sound deeper or more nasal.

In practical applications, formant shifting can be used to:

- Alter the perceived gender of a vocal performance.
- Create robotic vocal effects.
- Add harmonic layers to a vocal by duplicating the original audio and applying formant shifting.

LFO formant modulation can add rhythmic variations to a vocal, while confidence signals can adjust the formant shifts based on the intensity or stability of the beat. In Figure 3.8, the effect of periodic formant shifting applied to a demo audio track is visualized.

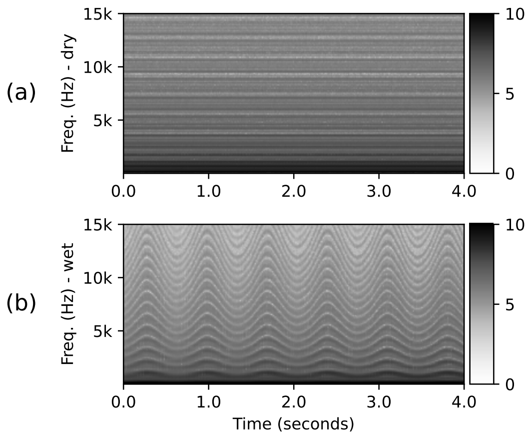


Figure 3.8. (a) Dry signal without any formant shifting, characterized by its original harmonic content. (b) Wet signal with periodic formant shifting applied, demonstrating changes in the harmonic structure and timbre.

The dry signal, shown in 3.8a, represents the original audio without any formant shifting applied. The wet signal, displayed in 3.8b, shows the audio with a periodic formant shifting effect applied. The signal alternates between its original dry state and the added formant harmonics, which are visible as a wave-like pattern in the spectrogram. An applied example of using periodic formant shifting with LFO signals is provided in case study 4.2.2.

3.5.2 Auto Filter

An auto filter can be used to modulate the frequency range of a sound. When applied as a highpass filter, the auto filter removes lower frequencies, resulting in a thinner, more treble-focused sound. Conversely, when used as a lowpass filter it removes higher frequencies, creating a warm and bass-focused effect. By using LFO or confidence signals, the filter cutoff can be dynamically adjusted to create sweeping or rhythmic filtering effects.

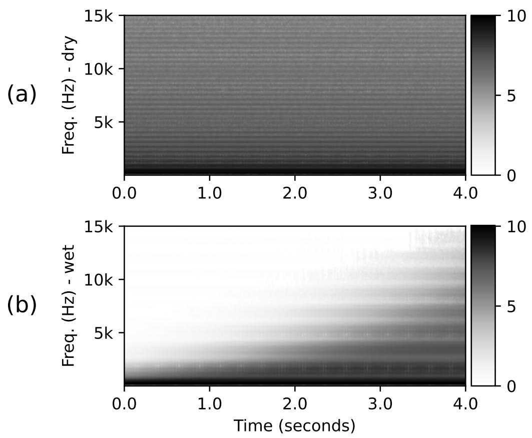


Figure 3.9. Spectrogram of an audio signal processed by a lowpass filter with a linearly increasing cutoff frequency. (a) Dry signal, characterized by a consistent frequency content across the entire spectrum. (b) Wet signal. The lowpass filter gradually allows higher frequencies over time.

In Figure 3.9, the effect of a lowpass filter with a linearly increasing cutoff frequency is illustrated. The signal in Figure 3.9a maintains a consistent frequency content across the entire spectrum, indicating no applied filtering effect. In music production this is called a "dry" signal. In contrast, the signal in Figure 3.9b shows how the higher frequencies are progressively added over time. In music production this is called a "wet" signal. The gradual inclusion of higher frequencies results in a brighter sound as the filter cutoff frequency increases.

3.6 Spatial and Time-Based Effects

In contrast to the frequency domain effects discussed above, this section will explore spatial- and time-based effects. The characteristics of an audio signal are manipulated to create a sense of width or space. The effects described in this section rest upon the methodologies presented in [8], [10], [12] and [20].

3.6.1 Reverb

Reverb is an effect that simulates the reflections of sound in a physical space, creating a sense of depth and ambience. Figure 3.10 illustrates the impact of reverb on a clap sound. The dry signal shown in 3.10a is a short, sharp burst of energy across a wide frequency range. The wet signal pictured in 3.10b includes the reverb effect, which adds a gradual decay in the frequency spectrum over time, resulting in a longer duration due to the reverb tail. By adjusting parameters such as room size, decay time, and the wet/dry mix, producers can place sounds in different virtual environments from rooms to cathedral halls. Reverb can be mixed using confidence signals to dynamically change the spatial characteristics of the sound based on different parts of a song.

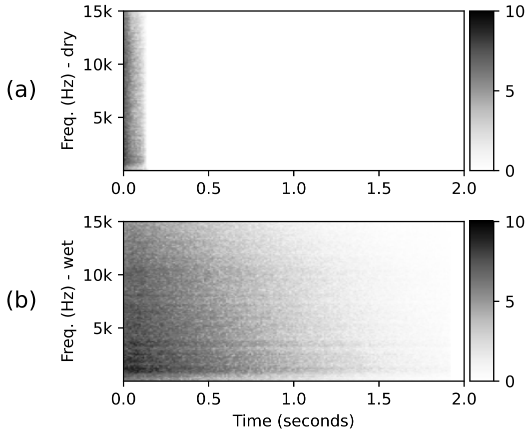


Figure 3.10. Spectrogram of a clap sound: (a) shows the dry signal without any reverb, characterized by a short, sharp burst of energy across a wide frequency range. (b) shows the wet signal with reverb applied, picturing the decay in the frequency spectrum over time, indicated by the added reverb tail. The wet signal has a longer duration, which comes from the effect of the reverb.

Another application of reverb is the so-called "reverb ducking". This procedure involves reducing the level of the reverb effect when a prominent sound, such as a vocal or kick drum, is present. This technique helps maintaining clarity in the mix by preventing the reverb from playing at the same time as important elements. Ducking can be controlled using LFOs to modulate the reverb level dynamically as described in 3.3.

3.6.2 Delay

Delay is an effect that records an audio signal and plays it back after a specified time interval, creating echoes. Delay effects can be synchronized with the beat by using LFO or confidence signals to modulate the delay time and feedback parameters, creating rhythmic echoes and repetitions. Figure 3.11 demonstrates the effect of delay applied to a demo audio track. The dry signal in Figure 3.11a represents the original audio without any delay effect. The wet signal shown in Figure 3.11b includes the delay effect, showing the echoes created by the repeated playback of the audio signal over time. The periodic repetitions in the wet signal are visible as vertical stripes in the spectrogram, corresponding to the delayed audio segments. Similar to the

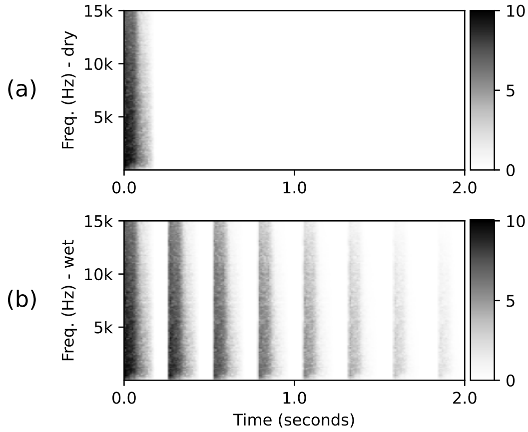


Figure 3.11. Spectrogram of a clap sound: (a) Dry audio without any delay effect. (b) Wet signal including the delay effect, showing the echoes created by the repeated playback of the audio signal over time.

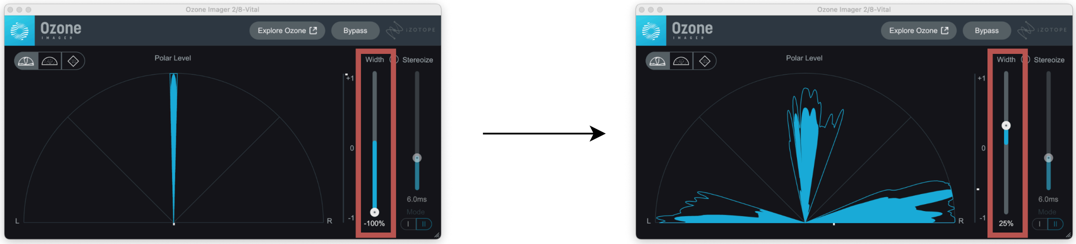
reverb effect outlined in Section 4.2.4, the delay effect can also be mixed into an audio track based on confidence control signals. By modulating the delay time and feedback parameters according to the confidence level, the delay effect can be used to create varying levels of echo patterns.

3.6.3 Panning

Panning effects can be dynamically controlled using LFO signals to create a sense of movement within the stereo field, enhancing the spatial dynamics of the track. A suitable audio track would be a shaker that is panned from left to right in sync to the beat. Also, a vocal with an already applied stutter effect can be further enhanced in the stereo field by adding panning based on LFO control signals.

3.6.4 Stereo Width Mixing

A mono audio signal creates a less immersive sound experience than an audio signal with stereo characteristics. The stereo width of an audio can be dynamically controlled using confidence signals from the beat tracker to create a more spatially dynamic and wider sounding mix. An example is given in Figure 3.12, where the Ozone Imager plugin is shown [4]. It allows to visualize and control the stereo width of an audio signal. The central vertical line represents the mono signal, where the left and right channels are identical. The horizontal spread indicates the stereo width. A wider spread means a wider stereo image, while a narrow spread means a narrower or more mono-centric image. Marked in red is the width control parameter. A demonstration of a



(a) (b)

Figure 3.12. Screenshot of the Ozone Imager plugin [4]. (a) Mono audio signal, width parameter set to -100%. (b) Enhanced stereo audio signal, width parameter set to $+25\%$.

mono audio signal is presented in 3.12a, created by setting the width parameter to -100%. The same audio signal can be enhanced in the stereo field like shown in 3.12b by setting the width parameter to $+25\%$.

Chapter 4

Creative Music Production

In this chapter, the integration possibilities of an online beat tracker concept in DAWs used for music production and mixing are explored. A demo track was produced to illustrate the practical application of the beat tracker effects, from which several case studies will be presented. These case studies highlight different sub-categorized application scenarios and are described and evaluated in detail. Pre-generated control signals are used to ensure reproducibility. The case studies, along with their theoretical descriptions, are accompanied by Jupyter Notebooks. These notebooks, in the form of HTML documents, include exported audio examples and the code for the plots given in this thesis.

4.1 Integration Possibilities in Creative Music Production

In this section, the integration of the beat tracker in digital audio workstations (DAWs) for various audio effect and mixing scenarios is outlined. The flowchart in Figure 4.1 illustrates the

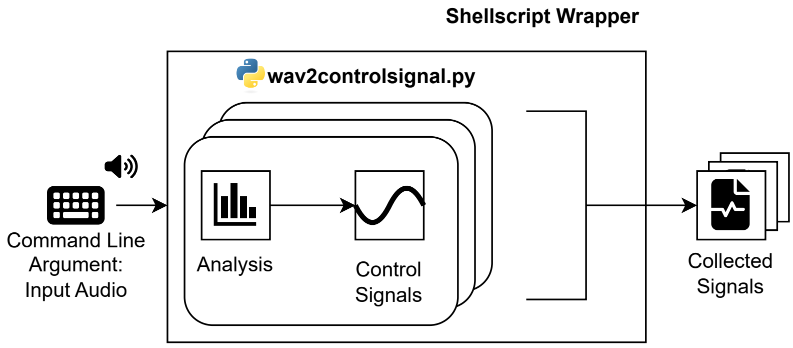


Figure 4.1. Flowchart for the generative process of control signals with varying configuration settings.

process of generating control signals with varying configuration settings. This generative process involves using different audio inputs and settings to produce control signals that can then be used in the case studies to modulate various parameters in the DAW. The goal is to generate LFO and confidence-based signals using multiple parameter settings. These settings include different tempi, a variety of tempo ranges to generate multiple tempo octaves, and different drum inputs. For this purpose, a shell script was developed to automate the process of generation for different examples, which can be found in Appendix C.1. After the generation, the control signals in form of `.wav` files are sorted in folders based on parameter settings and then imported into the DAW.

4.2 Demo Track and Case Studies

The goal of this section is to demonstrate how the control signals generated by the beat tracker can enhance the creative music production workflow. The following case studies are all part of one demo track, whose waveform and tempo in BPM over time are shown in Figure 4.2a and Figure 4.2b.

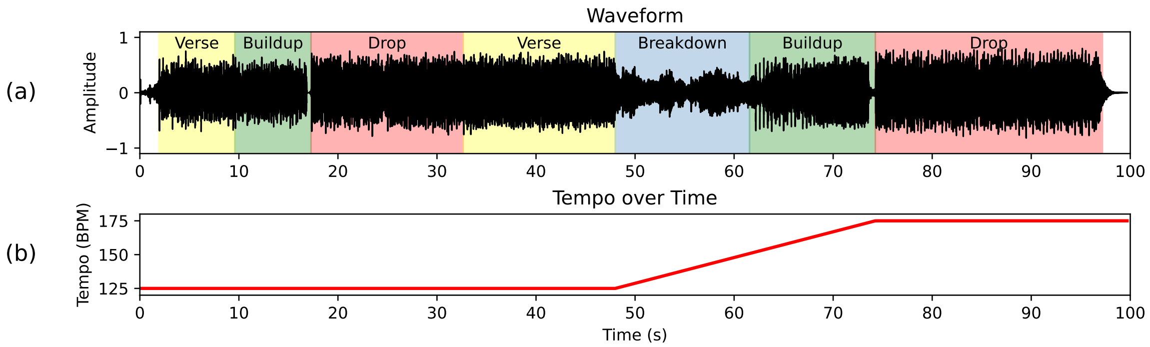


Figure 4.2. Waveform of the demo audio track (a) and its tempo in BPM over time (b). The demo track is divided into colored blocks representing different sections.

The demo track features different sections, represented by different highlighted colors in Figure 4.2a (which will be referred to later in the individual case studies), including various instrumentation and tempi.

1. **Verse:** The verse refers to the part of the song that tells a story or progresses the theme with lyrics and melodies.
2. **Buildup:** This section increases in intensity, gradually adding elements to create anticipation. It's designed to prepare the listener for the high-energy part of the track that follows.
3. **Drop:** The drop is the climax of the track where the energy peaks. It usually features a heavy beat and a catchy melody or bassline.

4. **Breakdown:** After the drop, the breakdown reduces the intensity, removing elements to create a contrast. It often provides a moment of relief before building back up to the next section.

The tempo structure of the demo track, pictured in 4.2b, can be explained as follows: The Verse in the start is at a tempo of 125 BPM with a “4-on-the-floor” beat during the first Drop and Verse sections. This is followed by a linear increase in tempo, reaching up to 170 BPM, incorporating drum patterns that build up tension. The closing section at 170 BPM features a Drum-and-Bass beat pattern. By including different tempi and a variation of genre from House to Drum-and-Bass music in the demo, different application scenarios of the beat tracker can be outlined.

Each case study explores different effects and processing techniques applied to individual elements within the demo track using the control signals generated by the beat tracker. By varying parameter settings, such as tempo ranges, tempo octaves, and drum inputs from different parts of the demo, the versatility of the beat tracker in a music production context is shown. To listen to the demo and individual case studies, refer to the Jupyter Notebook accompanying this thesis.

4.2.1 Arpeggiator

One main component of the demo track is its so called “pluck lead”. In music production, this refers to a synthesized instrument that produces a sharp, quick sound similar to plucking a string on a guitar or a harp. This sound is characterized by a fast attack and a short decay, meaning it reaches its peak volume quickly and fades away just as fast. In the demo track, it is introduced in the first drop, featuring a unique rhythm. The creation of rhythmic patterns in sync to the beat using an arpeggiator was described in 3.2. Based on this principle, in the following snippet of the demo track the rhythmic effect of a plucked lead instrument is described. The pipeline for creating this effect using the γ-LFO control signal and map it to the arpeggiator rate is pictured in Figure 4.3.

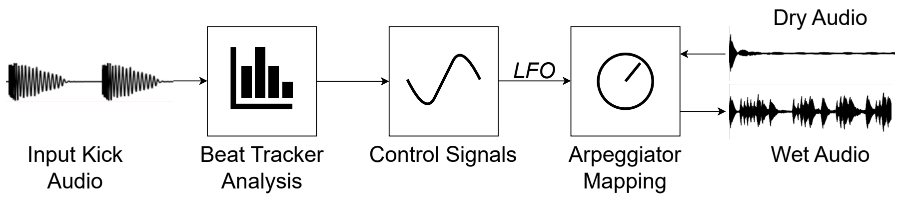


Figure 4.3. Pipeline: The kick input signal undergoes beat tracking analysis to extract beat information. Control signals are generated. The γ-LFO is mapped to the arpeggiator rate of the pluck instrument. The final output is the effected pluck lead instrument, arpeggiated in a rhythm in sync with the beat.

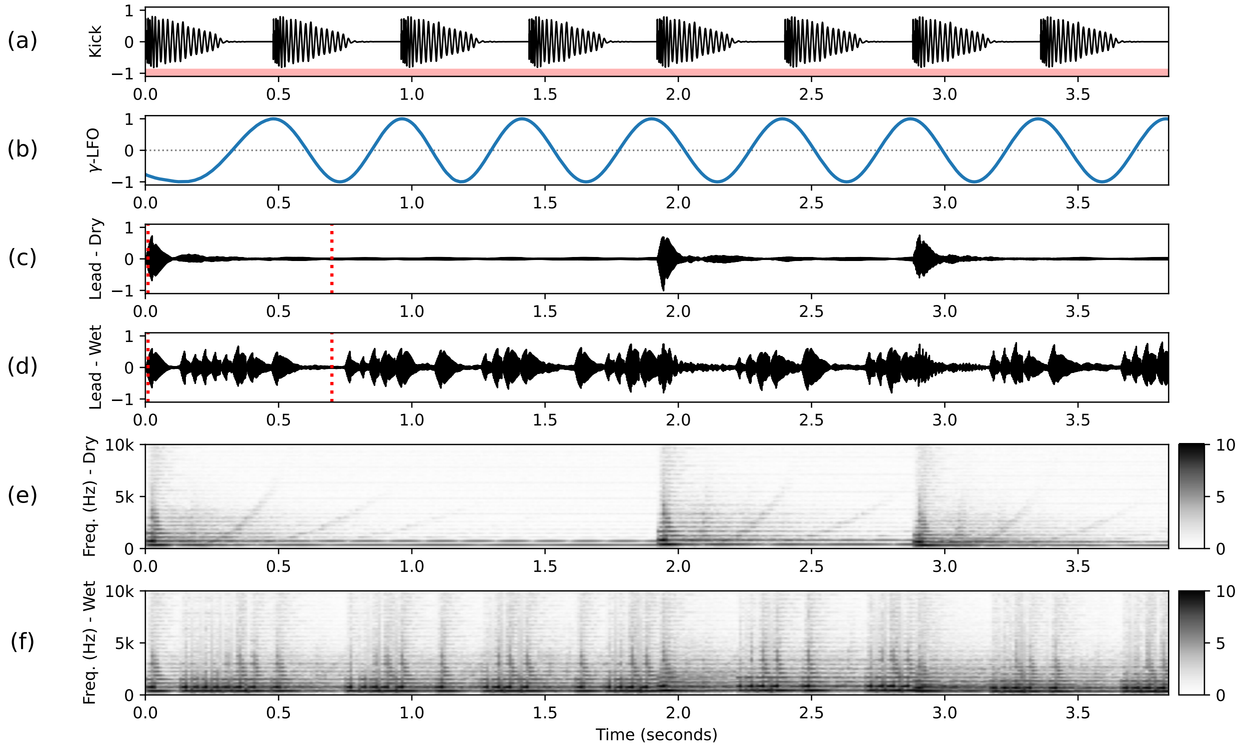


Figure 4.4. (a) 2-Bar 125 BPM loop of the kick drum. (b) Corresponding γ-LFO. (c) Dry pluck instrument. (d) Wet, arpeggiated pluck instrument. (e) Dry pluck spectrogram. (f) Wet, arpeggiated pluck spectrogram.

The lead pluck instrument is playing three notes in its dry variant, which can be observed in Figure 4.4c and 4.4e. The snippet pictured in Figure 4.4 refers to the red highlighted first drop part of the demo track shown in Figure 4.2, starting at bar 9 (17.28 s). To achieve the rhythmic variations in the wet lead sound, the following effect chain is applied: Whenever the LFO increases to its peak, the arpeggiator repeats the pluck sound at a faster pace. Conversely, when the LFO lowers, the pluck gets arpeggiated at a slower speed. The maximum and minimum values of the arpeggiator rate are limited: The mapping corresponds to the $\gamma-$LFO signal in a range from 28 % to 84 %. These values correspond to an arpeggiator rate of 36 ms and 480 ms.

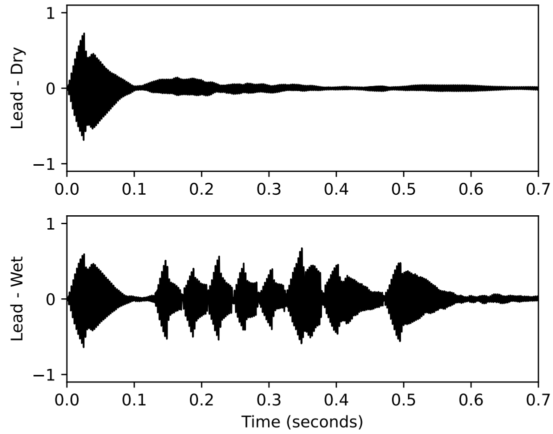


Figure 4.5. Close-up zoom of the dry and wet lead pluck instrument in comparison.

The contrast of the empty space in the dry lead pluck instrument compared to the effected, wet sound is visible in the waveform representations 4.4c and 4.4d as well as the spectrograms in 4.4e and 4.4f. Figure 4.5 pictures a closeup snippet of a time frame of 0.7 s starting at zero from Figure 4.4c and 4.4d, limited by red dotted lines. The rhythmic additions in the wet signal are distinctly noticeable.

4.2.2 Formant

During the breakdown of the demo track, formant manipulation is applied to the choir vocals, creating variations that sync with the linearly increasing tempo. As discussed in Chapter 3.5, formants are the frequency bands that define the tonal quality of a sound, particularly in human speech and singing.

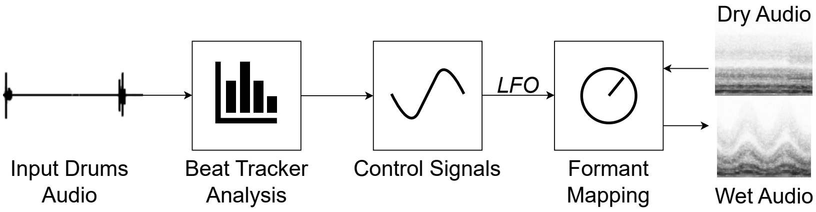


Figure 4.6. Pipeline: The kick input signal undergoes beat tracking analysis to extract beat information. Control signals are generated. The α-LFO is mapped to the formant on the audio. The final output is the effected vocal, modified in sync with the beat.

The pipeline for creating the effected vocals is shown in Figure 4.6. After control signal generation, the α-LFO is mapped to the formant control on the audio track to create the wet signal.

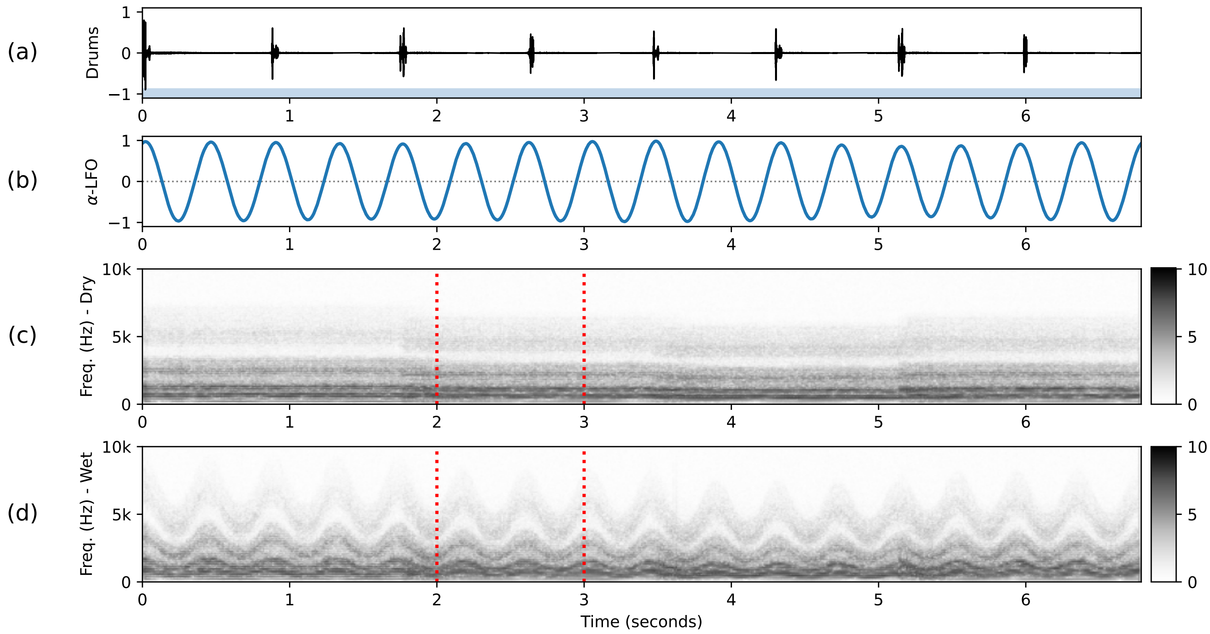


Figure 4.7. (a) 4-Bar loop of the drums. (b) Corresponding α-LFO. (c) Dry choir spectrogram. (d) Wet choir spectrogram.

This demo refers to the section starting at the time 55.36 s in the blue highlighted breakdown part of the demo track pictured in Figure 4.2. During this time, in the drums section only one percussive sample is playing with a linear tempo increase at a rate of 1/2 notes. The drums section was used to generate the α-LFO signal with the settings of a temporange between 120 BPM - 170 BPM, a 6s buffer and a lookahead of 0 s.

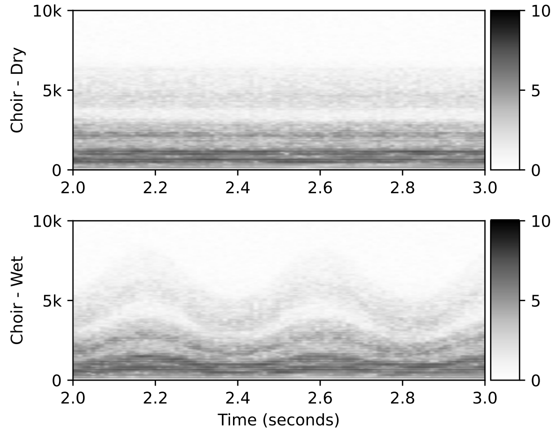


Figure 4.8. Closeup zoom into a short 1-second section of the waveform shown in Figures 4.7c and 4.7d. In the wet signal, the change in frequencies is clearly visible in comparison to the dry signal.

A closeup with duration of 1 s for the created effect is pictured in Figure 4.8. The snippet is marked by red dotted lines in Figure 4.7. To achieve the visible variations in the wet choir sound, the following effect chain is applied to the Auto Shift plugin in Ableton Live pictured in Figure 4.9. Whenever the LFO increases to its peak, the choir is transformed into a brighter tonality. Conversely, when the LFO falls the choirs tonality can be described as more dark and muffled. The maximum and minimum values of the formant are limited: The mapping of the α-LFO signal are in the range from 35 % to 65 %. These values correspond to a value of -29 % and $+29$ % in the formant setting. A value of 0 % represents the neutral, unaffected audio.

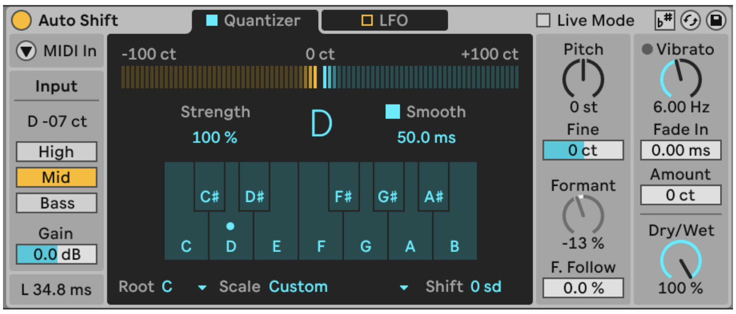


Figure 4.9. Auto Shift tuning plugin in Ableton Live, which was just recently added in the 12.1 beta version. It offers real-time pitch tracking, correction and various other vocal effects.

4.2.3 Tremolo - Vocal Stutter

In this case study, the application of a vocal stutter effect to a choir layered on top of an orchestral violin string section in the demo track is explored. Before the first drop at the beginning of the track, the choir vocals are played without any stutter effect, and the natural harmonics of the vocals can be heard. This creates a clear introduction to the track. To enhance the rhythmic impact during the drop, a tremolo or vocal stutter effect is applied to the choir vocals. This effect introduces rapid, rhythmic interruptions to the vocal signal, creating a percussive texture that complements the energy of the drop.

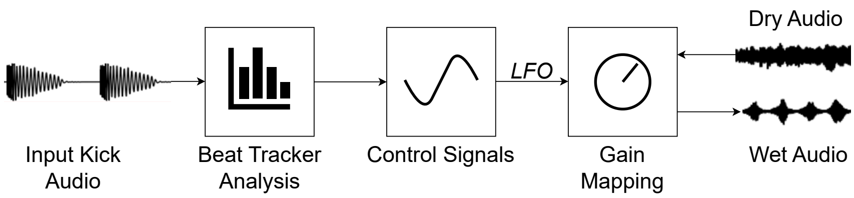


Figure 4.10. Pipeline: The kick input signal undergoes beat tracking analysis to extract beat information. Control signals are generated. The γ-LFO is mapped to the gain of the vocal. The final output is the vocal with a rhythmic stutter effect.

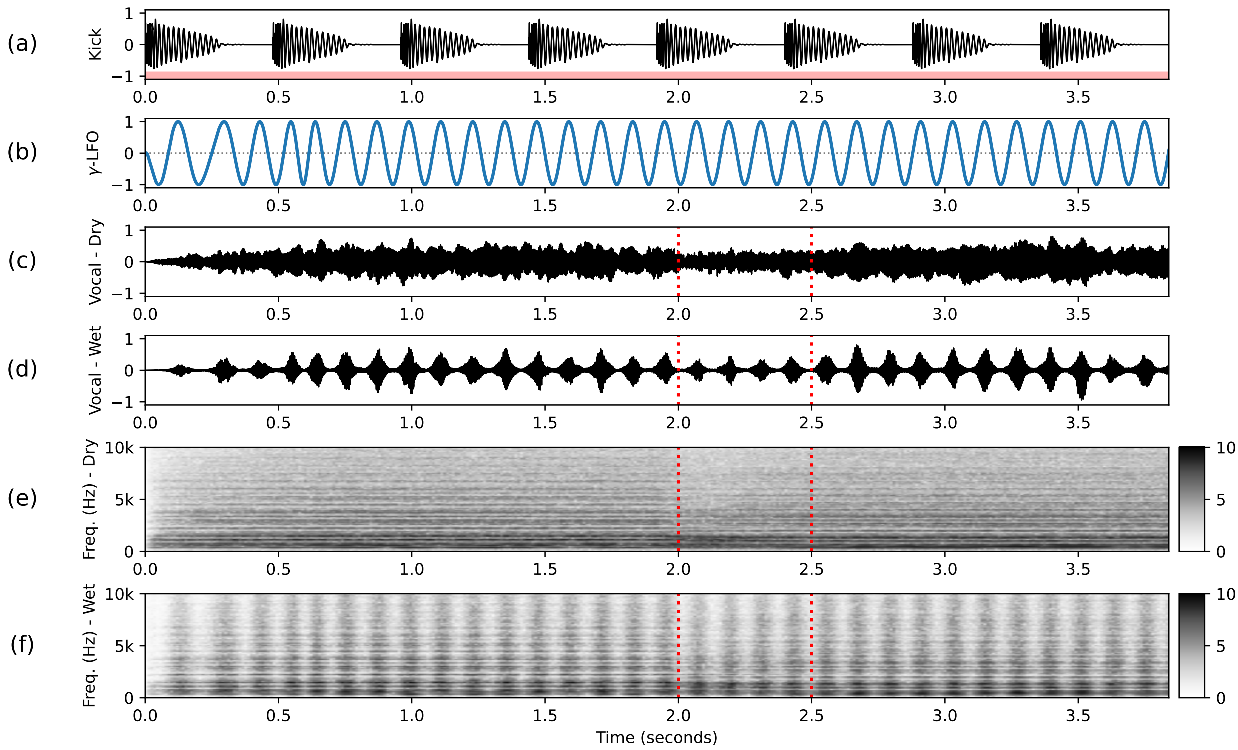


Figure 4.11. (a) 2-Bar 125 BPM loop of the kick drum. (b) Corresponding γ-LFO. (c) Dry vocal audio. (d) Wet vocal audio. (e) Dry vocal spectrogram. (f) Wet vocal spectrogram.

This demo refers to the red highlighted first drop part of the demo track pictured in Figure 4.2, starting at a time of 17.28 s. In Figure 4.11, the kick audio and its corresponding generated LFO, as well as the dry and wet signals are pictured. Clearly visible are the rythmic interruptions in the wet signal, created by the tremolo effect: A high LFO value corresponds to a time frame where the volume of the vocal is playing at its original level. The low LFO values correspond to an applied gain reduction. A close-up view of the dry and wet waveform as well as the dry and wet spectrograms is pictured in Figure 4.12. This short time frame of 0.5 s starting at a time of 2.0 s is marked in red dotted lines in the overall snippet pictured in Figure 4.11. It visualises the applied stuttering effect.

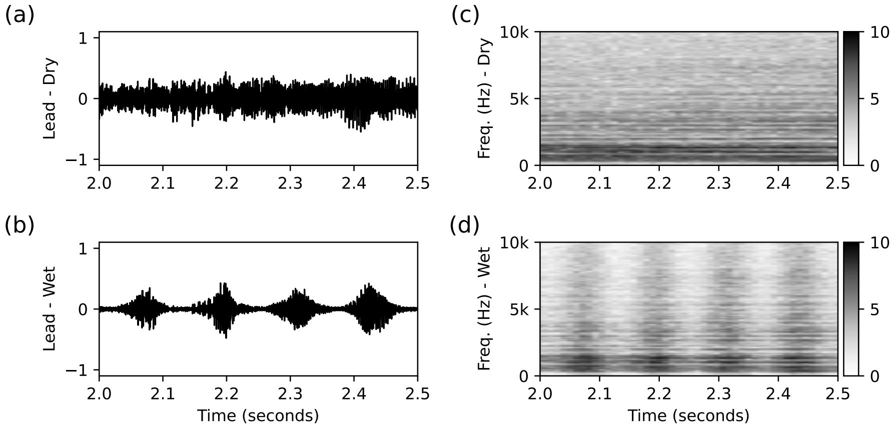


Figure 4.12. Zoomed-in view of the applied vocal stutter effect. (a) Waveform of the dry lead vocal signal. (b) Waveform of the wet vocal audio signal with the stutter effect. (c) Spectrogram of the dry lead vocal signal. (d) Spectrogram of the wet vocal with the stutter effect. Visible in the wet representations are the rhythmic interruptions in the waveform and corresponding modulations in the frequency content created by the stutter effect.

As pictured in the pipeline overview in Figure 4.10, the control signal was generated from the Kick input audio. During generation, a tempo range from 400 BPM to 560 BPM, a buffersize of 2 s and a lookahead value of 30 ms were used. In this scenario, these parameter settings are special. The BPM range is high, because the stutter rate is desired to be at the rate of 1/16 notes. This aligns with the fourth tempo octave of the actual tempo at the time, which is 125 BPM. The 30ms lookahead was chosen to create a subtle rhythmic variation. The value of 30 ms corresponds to a note length of 1/32 at 125 BPM. Even though the original purpose of the lookahead setting was to eliminate latency issues, it can also be used creatively. Common LFOs integrated in DAWs include a setting to shift the phase of the oscillation. A similar effect can be achieved using the lookahead parameter, which is visualized in Figure 4.13. By shifting the stuttering effect to be slightly off-beat, the overall rhythm is more dynamic. This technique highlights the creative potential of using lookahead adjustments in beat-synchronous audio effects.

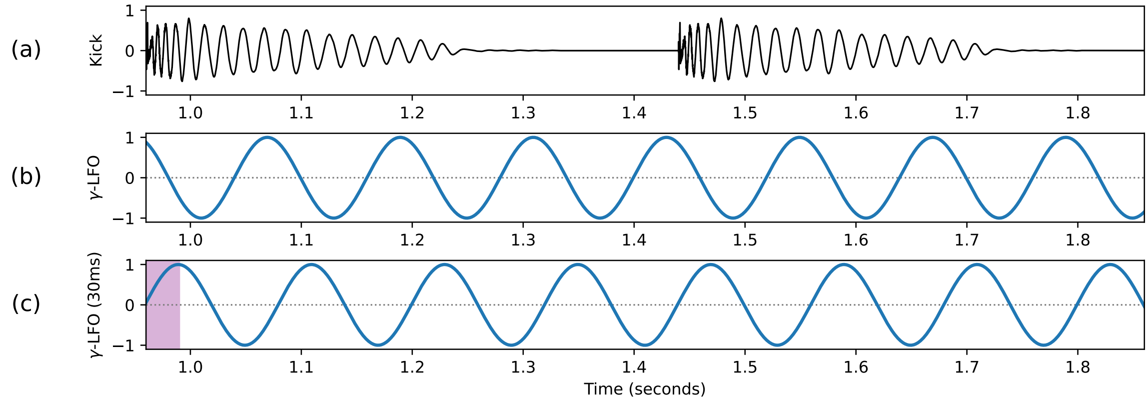


Figure 4.13. Comparison of parameter settings for control signal generation. (a) Kick input audio. (b) γ-LFO with lookahead parameter of 0ms. (c) γ-LFO with lookahead parameter of 30ms. The visible shift is highlighted in purple.

4.2.4 Reverb Ducking and Mixing

In this case study, the application of reverb ducking and mixing effects to the piano in the demo track is explored. Reverb is commonly used to create space and depth in audio production, but it can lead to interference with other instruments in the mix creating a "muddy" sound, as described in Chapter 3.6. Techniques to avoid this unwanted side effect are shown now.

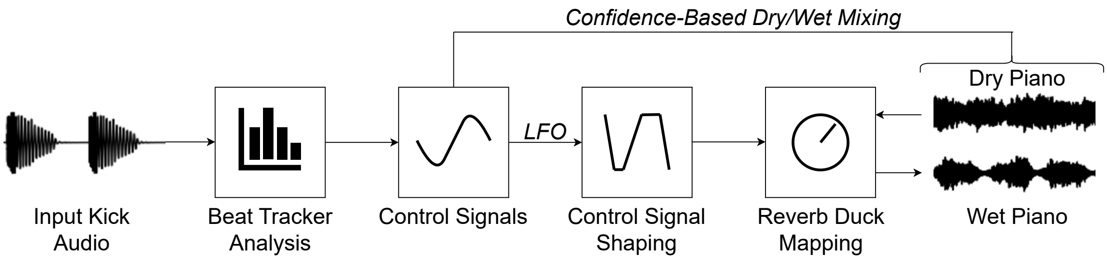


Figure 4.14. Pipeline: The kick input signal undergoes beat tracking analysis. Control signals are generated. The α-LFO is mapped to duck the reverb on the piano. The β-confidence signal is used to mix in the reverb ducking only during the drop.

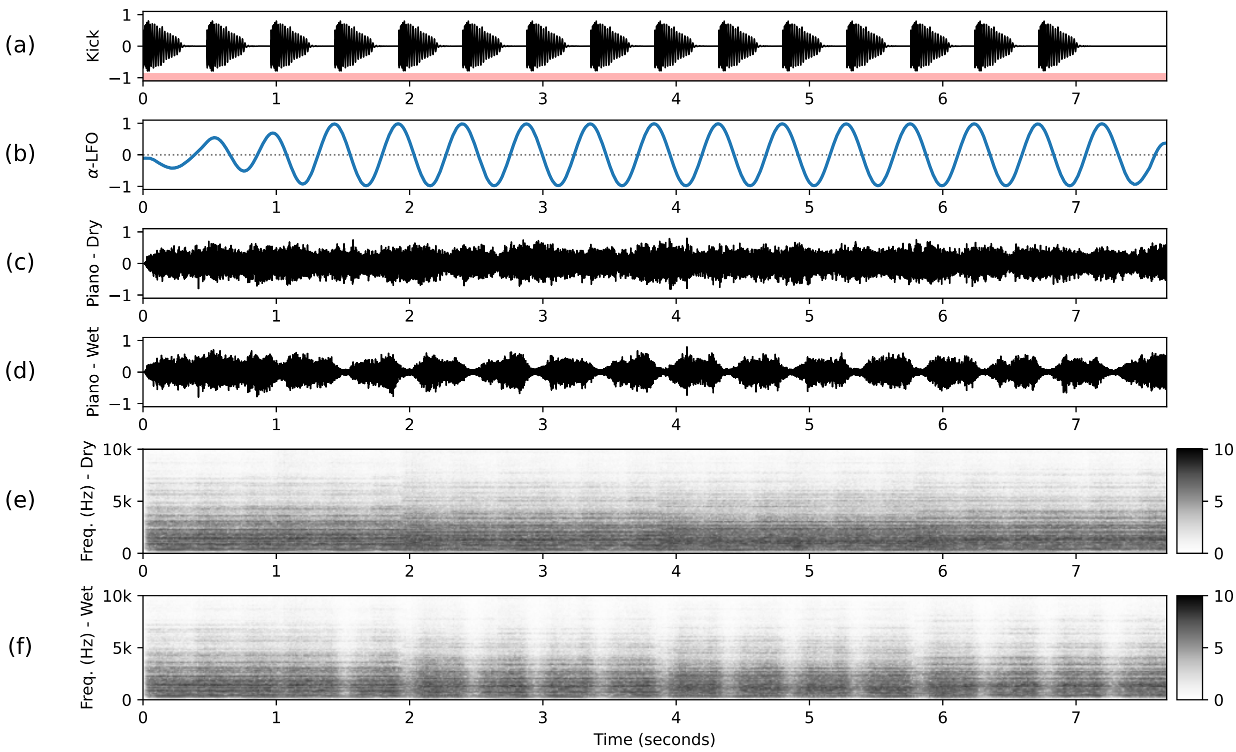


Figure 4.15. (a) 4-Bar 125 BPM loop of the kick drum. (b) Corresponding α-LFO. (c) Dry piano audio waveform. (d) Wet piano audio with visible reverb ducking applied. (e) Dry piano audio spectrogram. (f) Wet piano audio spectrogram with ducked reverb.

In the demo track, reverb ducking is used during the drop section to ensure the kick drum and lead instruments remain clear while still benefiting from the spatial effect of the reverb. This effect is shown in the audio snippet pictured in Figure 4.15. The audio snippet refers to the red highlighted first drop part of the demo track pictured in Figure 4.2, starting at time 17.28 s. As pictured in the pipeline presentation in Figure 4.14, the generated α-LFO control signal is used to control the ducking of the reverb. The control signals were generated using a tempo range of 100 BPM to 140 BPM, a buffersize of 2 s, and a lookahead of 0 ms. A high LFO value indicates a kick playing. At these time positions, the reverb effect is not relevant in the mix, and must be muted, which can be observed as light vertical lines in the spectrogram in 4.15f. Conversely, in periods of low LFO values the reverb effect is mixed in to create a richer and fuller sound.

An additional technique is to control the dry/wet mix of the reverb using confidence-based signals throughout the track, enhancing the overall depth. The audio snippet pictured in Figure 4.16 refers to the time 2 bars before the red highlighted first drop and is part of the demo track pictured in Figure 4.2. The parameter settings for the control signal generation are the same as above. For confidence-based applications of the reverb, a threshold is set at 82 % of the β-confidence level pictured in Figure 4.16c. This threshold is exceeded shortly after the kick starts playing at the red drop section shown in 4.16a because the α-LFO shown in 4.16a picks up the beat. Marked in Figure 4.16d in purple is the part of the mix where the reverb effect in 4.16e is included.

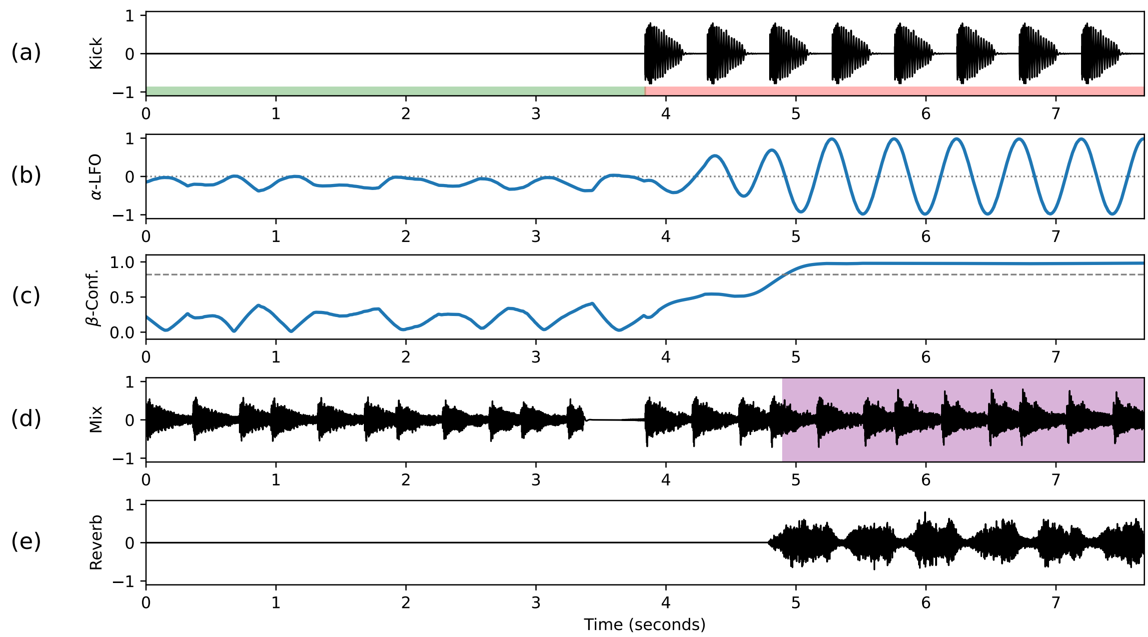


Figure 4.16. (a) 4-Bar 125 BPM loop of the kick drum. (b) Corresponding α-LFO. (c) Corresponding β-confidence. (d) Overall mix of the audio. Highlighted in the purple box is the part where the β-confidence crosses the set treshold, and thus the reverb effect is activated. (e) Isolated reverb of the piano.

4.2.5 Kick and Bass Sidechaining

With the debut of sidechain compression made popular by Daft Punk in the early 2000s, it is now widely spread in Dance music. The idea of sidechaining is to create room for other instruments while another sound is playing. A common example is to “duck” the volume of the bass sound while the kick is playing to avoid their frequencies from interfering. This results in the well-known pumping sound of the bass. By the traditional sidechaining approach a compressor is applied onto the bass signal channel. Whenever a specified volume threshold of the kick is received the gain is then ducked, resulting in the characteristic pumping sound.

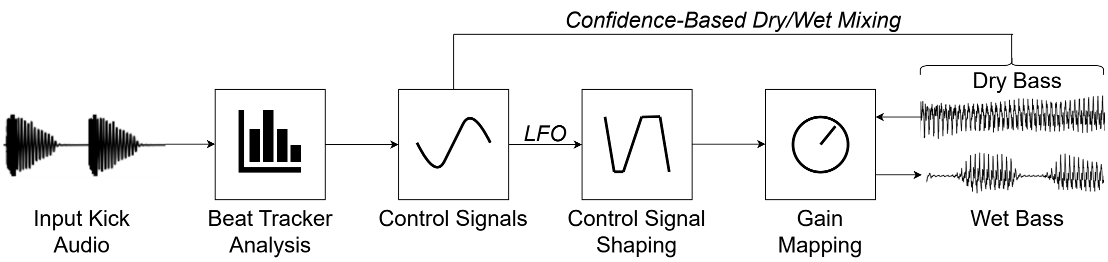


Figure 4.17. Pipeline: The kick input signal undergoes beat tracking analysis to extract beat information. Control signals are generated and then shaped. These modified signals are mapped to the gain control of the bass. The final output is the effected bass, synchronized with the detected beat.

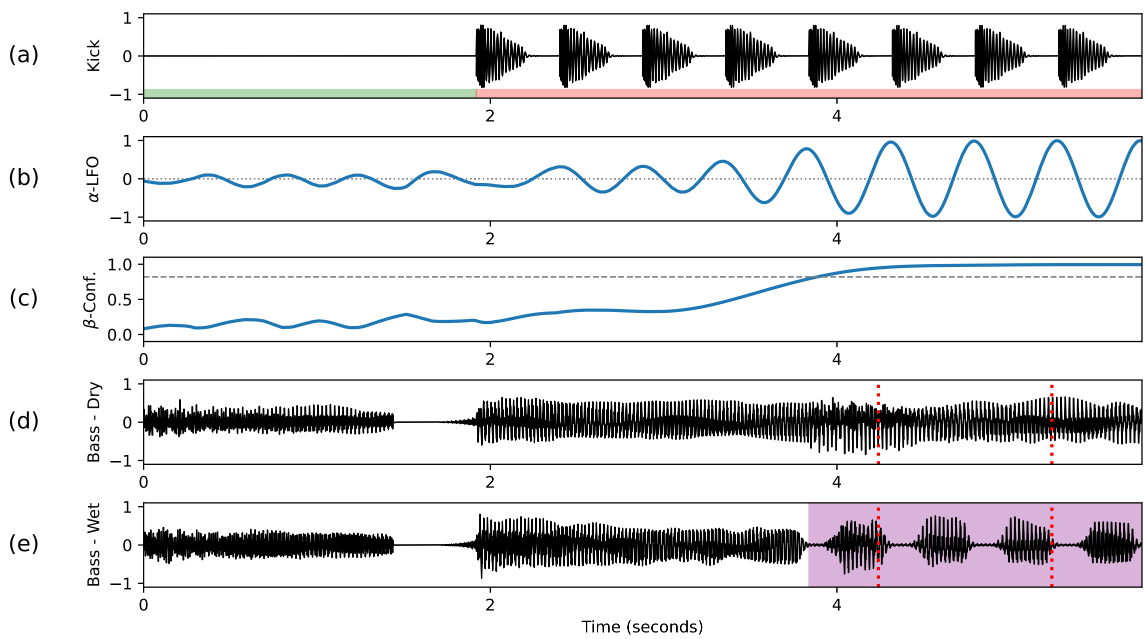


Figure 4.18. (a) 3-Bar 125 BPM loop of the kick drum. (b) Corresponding α-LFO. (c) Corresponding γ-confidence. (d) Dry bass audio. (e) Wet, sidechained Bass. Highlighted in the purple box is the part where the β-confidence crosses the set threshold, and thus the sidechain effect is activated.

A sidechaining effect on an example bassline can be created with the control curves from the real-time beat tracker and is illustrated in Figure 4.17. The parameter settings for generation were specified at a tempo range from of 100 BPM to 140 BPM, a buffer size of 6 s, and a lookahead of 0 ms. After control signal generation from the kick input audio, the α-LFO is mapped to the gain of the bass. The confidence signal is used with a threshold setting at 82 % to enable the sidechaining effect. This demo refers to the end of the first buildup in the demo track highlighted in green and transitions into the first drop section marked in red, as pictured in Figure 4.2 starting at second 21. The dry kick and bass loop are displayed in Figure 4.18a and Figure 4.18d. The bass is playing one note and the kickdrum is playing at the tactum level at 125 BPM. An extracted peak α-LFO value corresponds to a kickdrum hit, at this position the bass must duck its volume. The low values indicate time positions where room for the bass sound exists and the volume can be at its normal level. After the confidence threshold pictured in Figure 4.18c, the sidechaining effect is applied. The effected parts of the wet bass audio are shown in the purple box of Figure 4.18e. A closer comparison between the dry and wet bass signal is given in Figure 4.19. The closeup with duration of 1 s is taken from time 4.24 s, marked with red dotted lines in Figure 4.18d and Figure 4.18e.

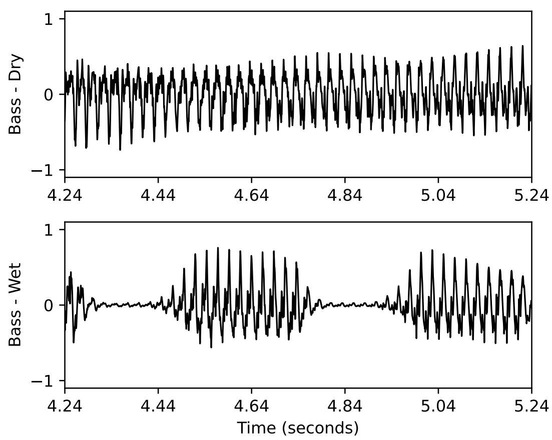


Figure 4.19. Closeup Zoom into a short time section of 1 second in the waveform of 4.18d and 4.18e. In the wet signal, the generated room is clearly visible in comparison to the dry signal.

A modification of the sinusoidal-shaped LFO signal is required before mapping it to the gain control of the bass sound. Sidechain patterns tend to only duck the sound right at the kick position and not symmetrically spaced like the sinusoidal pattern. The shaping of the signal can be accomplished with a customized Max 4 Live plugin in Ableton Live. For more details on the shaping process, see Chapter 3.1. As seen in Figure 4.20, the resulting shape is a stretched version of the α-LFO signal. After the control signal manipulation, the short ducks create room for the kick, while the stretched peaks give room for the volume of the bass at its high level.

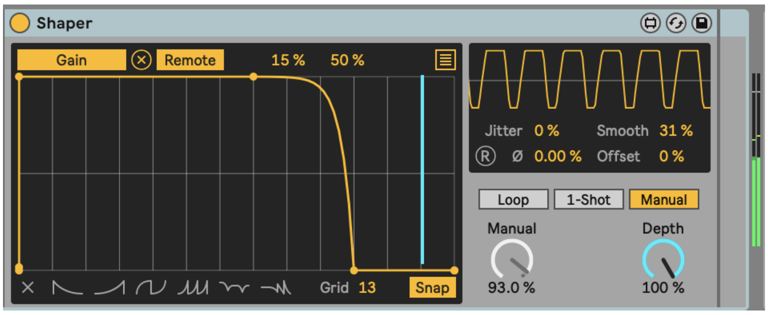


Figure 4.20. Shaper-plugin in Ableton Live with custom drawn input shape on the left, resulting in the output waveform pictured on the right, which can be mapped to the gain of the bass. The LFO control signal is mapped to the "Manual" regulator.

4.2.6 Wavetable Synthesis - Timbre

During the breakdown of the demo track, a blend between the genre of the first drop, House music at 125 BPM, and the genre of the second drop, Drum-and-Bass at 170 BPM, takes place. Both genres typically feature deep bass synthesizers. To introduce the listener to the tempo change, the bass is effected in sync with the drum track, which linearly increases its tempo. A wavetable blending effect, as described in Section 3.4, is applied here.

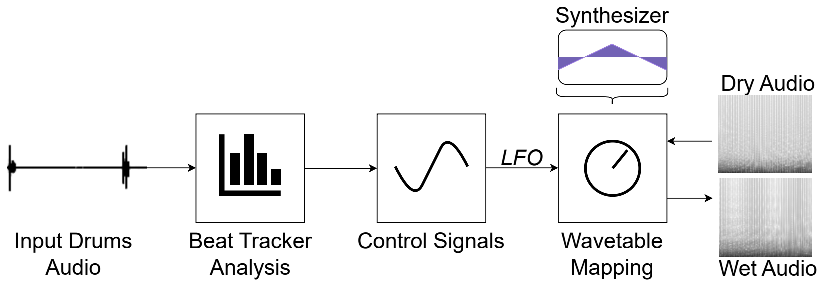


Figure 4.21. Pipeline: The drums input signal undergoes beat tracking analysis. Control signals are generated. The α-LFO is mapped to blend waveforms in the snythesizer. The final output is the effected bass synth.

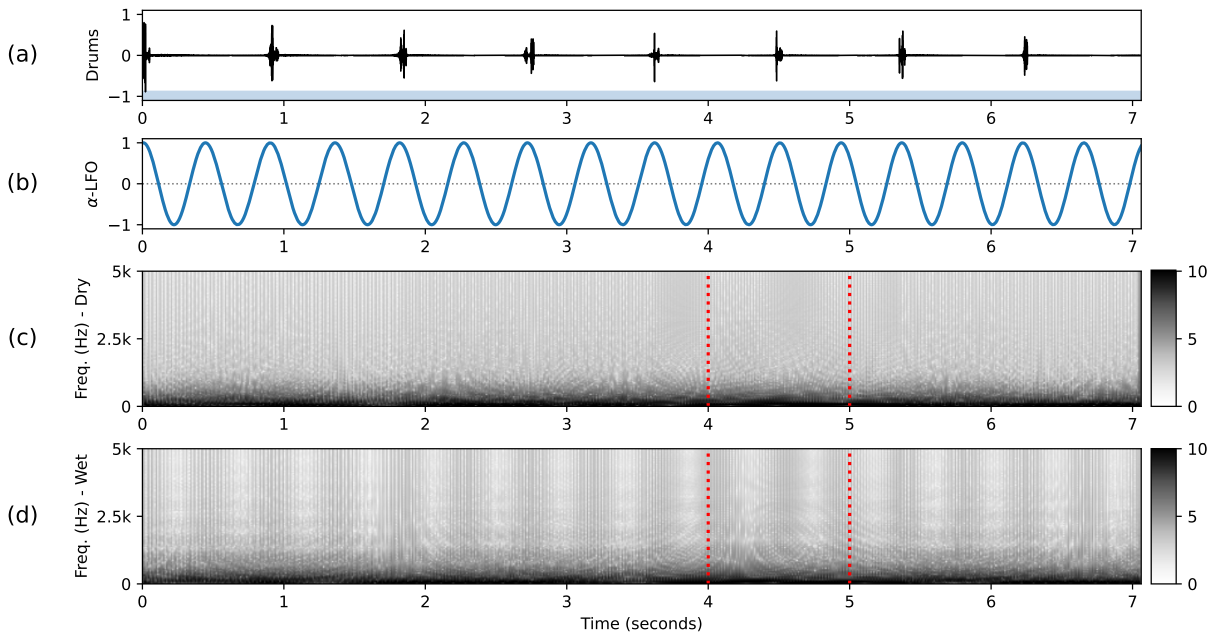


Figure 4.22. (a) 4-Bar loop of the drum section. (b) Corresponding α-LFO. (d) Dry bass synthesizer spectrogram. (e) Wet bass synthesizer spectrogram.

As shown in the pipeline presentation in Figure 4.21, the drums input audio is used to generate control signals. These were generated using a tempo range of 120 BPM to 170 BPM, a buffer size of 6 s, and a lookahead of 0 ms. This demo, pictured in Figure 4.22, refers to the blue highlighted breakdown part of the demo track shown in Figure 4.2, starting at 51.76 seconds. The drums α-LFO shown in Figure 4.22b is mapped to the Vital synthesizer plugin. Inside the synthesizer, the blending setting between two different waveforms is mapped, as pictured in Figure 3.6. A low LFO value corresponds to the original bass synthesizer setting. A high α-LFO value corresponds to a maximum mapping of 5 % in the synthesizer's blending setting.

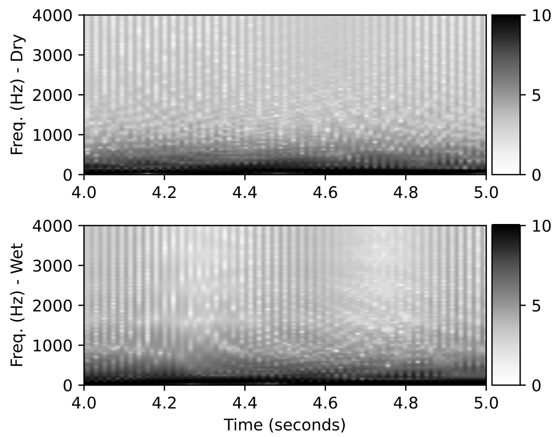


Figure 4.23. (a) Dry bass synthesizer spectrogram. (b) Wet bass synthesizer spectrogram.

In Figure 4.23, a 1-second closeup comparison starting at the time of 4 seconds in the dry (see 4.22a) and wet (see 4.22b) signals is provided. The time window is marked with red dotted lines in Figure 4.22. The blending setting of 5 % creates differences in the spectrogram that are minor but still observable. The effect can be listened to using the audio demo attached in the Jupyter Notebooks. The minimal parameter change creates a more distorted sound with new overtones. By subtly altering the timbre of the bass, the listener is introduced to the tempo change in a nuanced way. The drum section is sparse compared to other parts of the demo and would not energetically lead into the faster tempo parts without this effect in sync with the rhythm.

With the adaptation of control signal parameter settings during generation, these effects can be seamlessly applied to other parts of the song with diverse tempos, ensuring consistent and dynamic sound manipulation throughout the entire track.

Chapter 5

Conclusions

In this thesis, I demonstrated the potential of integrating real-time beat tracking into music production and mixing workflows. Theoretical foundations for real-time beat tracking were introduced in Chapter 2. Building on this, Chapter 3 presented LFO and confidence-based signal modifications, serving as a catalog for musicians and producers to explore various effects. One key discovery was the creative production of the demo track presented in Chapter 4, which relied on pre-generated control signals from an existing real-time beat tracker. This demo track features a variety of effects from the presented collection, based on tempo variations, showcasing the versatility of the new real-time beat tracker in handling different scenarios. The beat tracker effectively manages tempo changes and provides stable control signals, enabling flexibility in diverse music production contexts.

During the writing of this thesis, several new real-time software products, such as Mixed In Key Live and Logic Pro Session Players, were launched, underscoring the increasing demand for real-time music analysis and accompaniment. To further integrate the concepts and findings of this thesis into future projects or software applications, developing a more user-friendly interface and workflow for incorporating beat tracking and control signal generation into existing DAWs would be beneficial. This approach aligns with previous research, such as [18], and highlights the growing trend towards enhancing human creativity through intelligent systems [8]. These advancements underscore the relevance of this thesis for future developments in the field of real-time music production.

Appendix A

Glossar for Audio Terms

Ableton is a popular digital audio workstation (DAW) used for music production, mixing, mastering, and live performances [1].

ADSR (Attack, Decay, Sustain, Release) A model that describes the four stages of an envelope applied to a sound [12, 16, 19]. These stages shape the amplitude of the sound over time:

Attack The time it takes for the sound to reach its maximum amplitude after being triggered.

Decay The time it takes for the sound to decrease from the maximum amplitude to the sustain level.

Sustain The level at which the sound remains while the input is held.

Release The time it takes for the sound to decrease from the sustain level to silence after the input is released.

Audio Plugin Software that adds audio effects or virtual instruments to a DAW. Plugins can process audio or MIDI data and are used to enhance or alter sounds in music production.

Beatgrid Information Data represents the timing and spacing of beats in a music track, used for beat matching and tempo synchronization.

Beat Matching is the process of matching the tempo and phase of two or more tracks to ensure a seamless transition between them during a DJ set.

BPM (Beats Per Minute) A unit of measure that indicates the tempo of a piece of music.

DAW (Digital Audio Workstation) Software used for recording, editing, and producing audio files. Examples include Ableton Live, Ardour, Reaper, Pro Tools, Logic Pro and many more.

Dry Signal The unprocessed audio signal before any effects or processing are applied.

Envelope A curve describing the variation of a parameter over time, commonly used to shape the amplitude, filter, or pitch of a sound.

Equalization (EQ) The process of adjusting the balance between frequency components within an audio signal.

Frequency Clashing Occurs when two or more audio signals contain overlapping frequencies that cause interference, resulting in a muddled or harsh sound.

Lead In music production, "lead" refers to the primary musical line or instrument that carries the main melody or prominent rhythmic pattern in a track. Examples include the lead vocal, which is the main vocal track that delivers the primary lyrics or melody. Another example is the lead instrument, which is te main instrument in a section. This could be a guitar, synth, pluck, or any other instrument that stands out in the mix.

Mapping (of a control signal to another parameter) is the process of assigning a control signal, such as an LFO, to modulate a specific parameter within a DAW or plugin.

Mastering is the final step in music production that involves optimizing and preparing a track for distribution by adjusting levels, equalization, compression, and other audio enhancements.

Max 4 Live is a visual programming language within Ableton Live that allows users to program custom devices for audio and MIDI processing [2].

Mixed In Key is a software application used by DJs for assisted management of a song library. The program is used to analyze and detect the key, tempo, and other musical attributes of audio tracks [3].

Mixing is the process of combining multiple audio tracks into a single cohesive audio file by adjusting levels, panning, effects, and other parameters. A mixing engineer is a professional whose job it is to mix recordings for artists [10, 11].

MIDI (Musical Instrument Digital Interface) A technical standard that describes a protocol, digital interface, and connectors for connecting various electronic musical instruments, computers, and related devices [12, 16, 19].

Oscillator An electronic component that generates a periodic, oscillating electronic signal, often used in synthesizers to generate sounds.

Rekordbox is the DJ software by Pioneer that allows DJs to prepare and manage their music libraries, including track analysis for beatgrid and key information. The application also includes a live performance mode [5].

Song Structure The arrangement of different sections of a song that provides a framework for the musical composition, listed in the typical order of occurrence:

- **Verse** The section of a song that typically features the main lyrics and melody, providing the narrative or story.
- **Buildup** A gradual increase in intensity, leading to a drop.
- **Drop or Chorus** The most energetic part of a song, often in electronic dance music, where the rhythm and bassline are introduced.
- **Breakdown** A section where the music is reduced to fewer elements, creating a contrast to energetic parts in the song structure.

Timbre The characteristic quality of a sound that distinguishes it from other sounds of the same pitch and volume. Timbre is determined by the harmonic content of a sound and the dynamic characteristics of the sound, such as its attack, decay, sustain, and release (ADSR) envelope. It is often referred to as the “color” or “tone” of a sound.

Wet Signal The processed audio signal after effects or processing have been applied.

Appendix B

Additional Case Studies

These additional case studies highlight the versatility of the beat tracker and its potential applications in creative music production. Although they were not included in the demo track, they demonstrate the wide range of possibilities for using control signals to modulate audio effects dynamically.

B.1 Slink Devices Collection

The LFOs used in this thesis were generated by a new real-time beat tracker. To provide an outlook on other creative LFO possibilities, the Slink Devices collection [6] offers a variety of modulation opportunities. Pictured in Figure B.1 is a screenshot that displays the user interface of the "Looping Vibration CV" device from the Slink Devices collection within the Ableton Live DAW. This plugin is designed to create complex LFO patterns that can be mappet to various parameters in a digital audio workstation (DAW). The Slink Devices collection for the Ableton Live DAW includes different options for shaping and customizing LFO signals in creative ways. Complex and evolving modulation patterns can be created, demonstrating the potential for creative modulation using LFOs in music production.

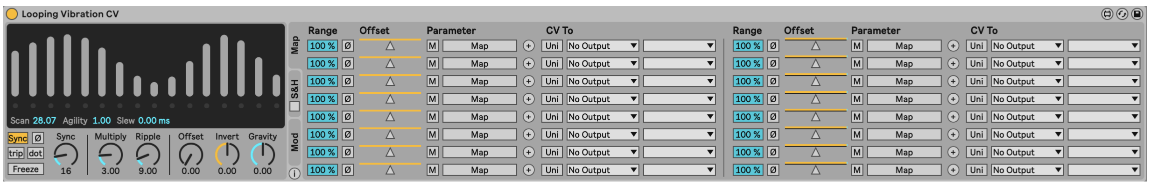


Figure B.1. Screenshot of the Slink LFO plugin with the "Looping Vibration CV" preset from the Slink Devices collection in Ableton Live. The interface includes a waveform display and control knobs for adjusting LFO characteristics (left), as well as a mapping section for routing modulation signals to different outputs (right).

B.2 Auto Filter

Within digital audio workstations and during live DJ performances, a commonly used effect is manipulating frequencies using a low- or high-pass filter. In this case study, a synthesized bass playing a single note is shaped in a rhythmically interesting way by modulating the cut-off frequency of an autofilter plugin, used as a low-pass filter. The bass shifts between a damped sound with only low frequencies passing through and a bright sound as the cut-off frequency of the filter increases.

With the desired effect of ducking the high frequencies of the bass whenever a kick is played, the control signals are generated from the "4-on-the-floor" drum pattern playing at 120 BPM (a). The resulting α-LFO (b) is used to modulate the cut-off frequency of the filter. A high α-LFO correlates to the kick and beat pattern, where the bass high frequencies shall be filtered.

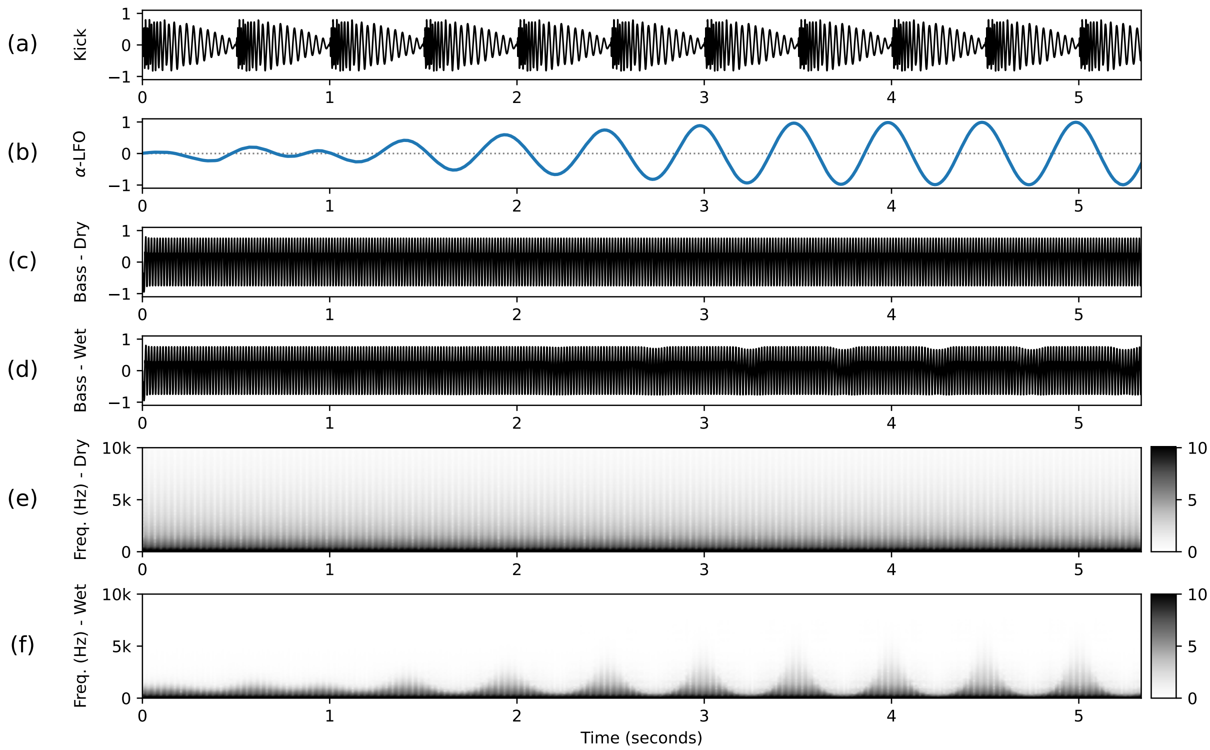


Figure B.2. 8-bar 120 BPM loops of the kick drum, its corresponding α-LFO, the affected bass wavetable and its spectrogram.

B.3 Kick and Bass Sidechaining - Additional Example

In this example, a bass playing 1/8 notes is sidechained with a kick.

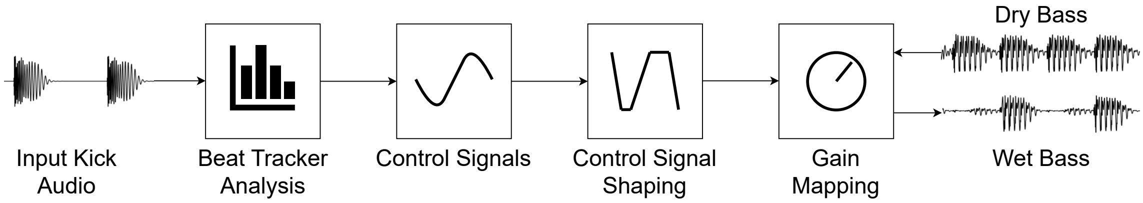


Figure B.3. Pipeline: The bass input signal undergoes beat tracking analysis to extract beat and tempo information. Control signals are generated and then shaped. These modified signals are mapped to the gain control of the bass. The final output is the effected bass, synchronized with the detected beat.

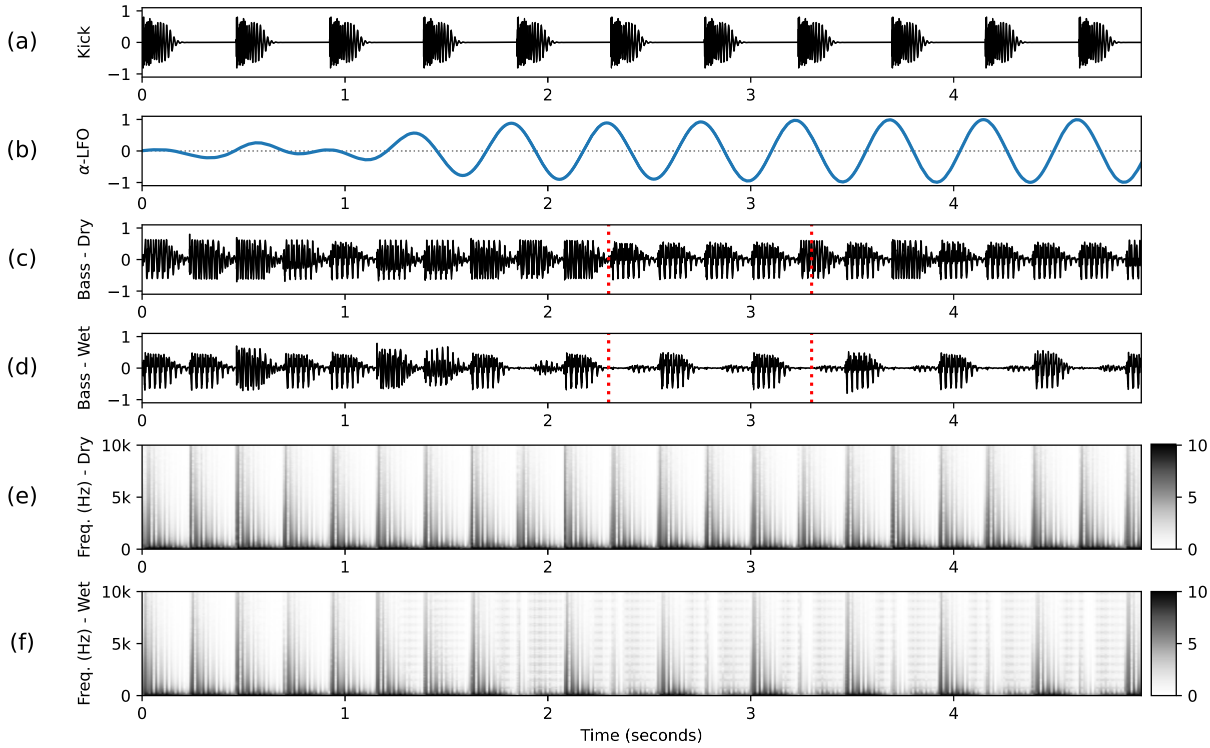


Figure B.4. (a) 8-bar 130 BPM loop of the kick drum. (b) Corresponding α-LFO. (c) Dry bass wavetable. (d) Wet bass wavetable. (e) Dry bass spectrogram. (f) Wet Bass spectrogram.

A sidechaining effect on an example bassline can be created with the control curves from the real-time beat tracker and is illustrated in Figure B.3. The dry kick and bass loop are displayed in Figure B.4a and Figure B.4c. The bass is playing in an eights-note pattern and the kick drum is playing at the tactum level at 130 BPM. A peak α-LFO value indicates a kick drum hit, and at this position, the bass must reduce its volume. The lower α-LFO values indicate times when the bass sound can play at its normal level because the kick drum is not playing, allowing room for the bass sound. A modification of the sinusoidal shaped LFO-signal is required before mapping it to the gain control of the bass sound. Sidechain patterns tend to only duck the sound right at the kick position and not symmetrically spaced like a the sinusoidal pattern.

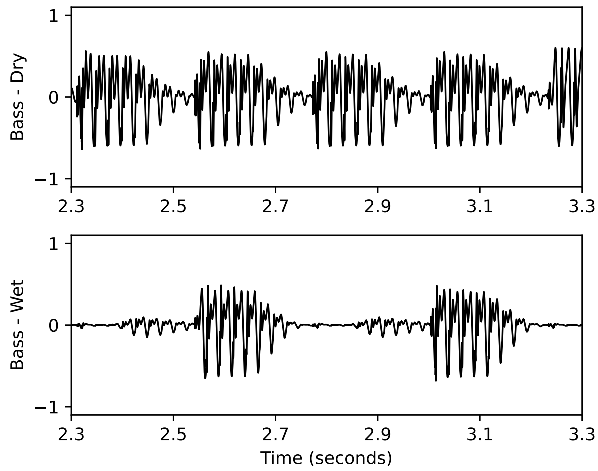


Figure B.5. Closeup Zoom into a short time section of 1 second in the waveform of B.4c and B.4d. In the wet signal, the generated room is clearly visible in comparison to the dry signal.

The shaping of the signal can be accomplished with a customized Max 4 Live plugin in Ableton Live. For more details to the shaping process, see Chapter 3.1. As seen in Figure B.6, the resulting shape is a stretched version of the α-LFO signal. After the control signal manipulation, the short ducks create room for the kick, while the stretched peaks give room for the volume of the bass at its high level.

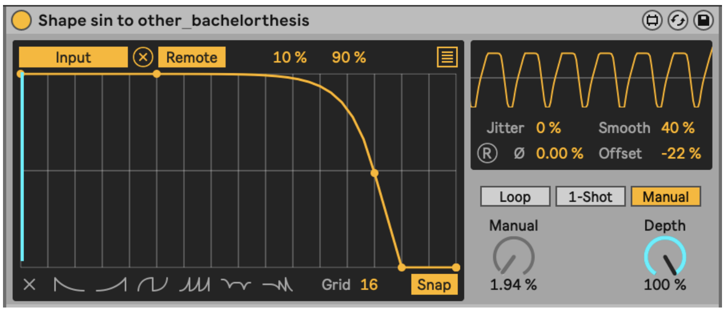


Figure B.6. Shaper-plugin in Ableton Live with custom drawn input shape on the left, resulting in the output waveform pictured on the right, which can be mapped to the gain of the bass.

Appendix C

Source Code

C.1 Shellscript wrapper

The `beatlib_wrapper.sh` shellscript is used as a wrapper for `wav2controlsignal.py`, provided by Peter Meier from an internal beatlib repository. The wrapper generates a collection of control signals with varying parameter settings in separated folders. The control signal files are later imported into the DAW Ableton Live.

Usage:

```
./beatlib_wrapper.sh <input_file>;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

#!/bin/bash
################################################################################
# shell-script to create control signals with different settings.
# This is made possible with Peter Meier's beatlib library.
################################################################################

# Check input
if [ -z "$1" ]; then
  echo "Usage: $0 <input_file>"
  exit 1
fi

input_file="$1"
input_file_type=${input_file: -3} # Get filetype
echo "Input file type: $input_file_type"
input_file_base=$(basename "$input_file" .$input_file_type) # Get base name
echo "Input file: $input_file_base"
input_file_dir=$(dirname "$input_file") # Getinput directory

# virtual env. Adjust to your machine!
venv_dir="YourPathToBeatlib/beatlib/.venv"
```

```
# Activate virtual env
source "$venv_dir/bin/activate"

# Parameter settings
buffersize_values=(512)            # default - 512
tempo_pairs=(
  "100 140"
  "120 170"
  "201 400"
  "400 560"
)
lookahead_values=(0)               # default -   0
kernel_values=(2 6)                # default -   6

# Create output dirs and run wav2controlsignals with different parameters
for buffersize in "${buffersize_values[@]}"; do
  for tempo_pair in "${tempo_pairs[@]}"; do
    tempo_low=$(echo $tempo_pair | cut -d ' ' -f 1)
    tempo_high=$(echo $tempo_pair | cut -d ' ' -f 2)
    for lookahead in "${lookahead_values[@]}"; do
      for kernel in "${kernel_values[@]}"; do
        # Naming for output dir
        output_dir="${input_file_dir}/ctrl_sig_${input_file_base}
            /output_buffersize_${buffersize}_tempo_${tempo_low}-${tempo_high}_lookahead_${lookahead}_kernel_${kernel}"
        mkdir -p "$output_dir" # create output dir

        echo "Generating sign with parameters:
            buffersize=$buffersize, tempo=$tempo_low-$tempo_high, lookahead=$lookahead, kernel=$kernel"
        python3 "YourPathToBeatlib/beatlib/scripts/wav2controlsignals.py" -i
            "$input_file" --buffersize "$buffersize" --tempo "$tempo_low" "$tempo_high" --lookahead
            "$lookahead" --kernel "$kernel" > "$output_dir/output.log"

        #move generated files (dependant on filetype)
        mv "$input_file_dir/${input_file_base}_alpha_lfo.${input_file_type}"
            "$output_dir/${input_file_base}_alpha_lfo_${tempo_low}-${tempo_high}_${kernel}.${input_file_type}"
        mv "$input_file_dir/${input_file_base}_beta_conf.${input_file_type}"
            "$output_dir/${input_file_base}_beta_conf_${tempo_low}-${tempo_high}_${kernel}.${input_file_type}"
        mv "$input_file_dir/${input_file_base}_gamma_conf.${input_file_type}"
            "$output_dir/${input_file_base}_gamma_conf_${tempo_low}-${tempo_high}_${kernel}.${input_file_type}"
        mv "$input_file_dir/${input_file_base}_gamma_lfo.${input_file_type}"
            "$output_dir/${input_file_base}_gamma_lfo_${tempo_low}-${tempo_high}_${kernel}.${input_file_type}"
        echo "Files moved to $output_dir "
      done
    done
  done
done

# Deactivate virtual env
deactivate
echo "Complete."

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Appendix D

Ableton Live Demo Track Project

The demo track was produced in the DAW Ableton Live, Beta version v.12.1b4 on a MacBook Pro 2021 Edition, with the Apple M1 Pro CPU and 16GB of RAM. The project lead the PC to CPU throttling due to the amount of effects used in the 96 audio and MIDI tracks. A screenshot of the overall project is attached in Figure D.1.

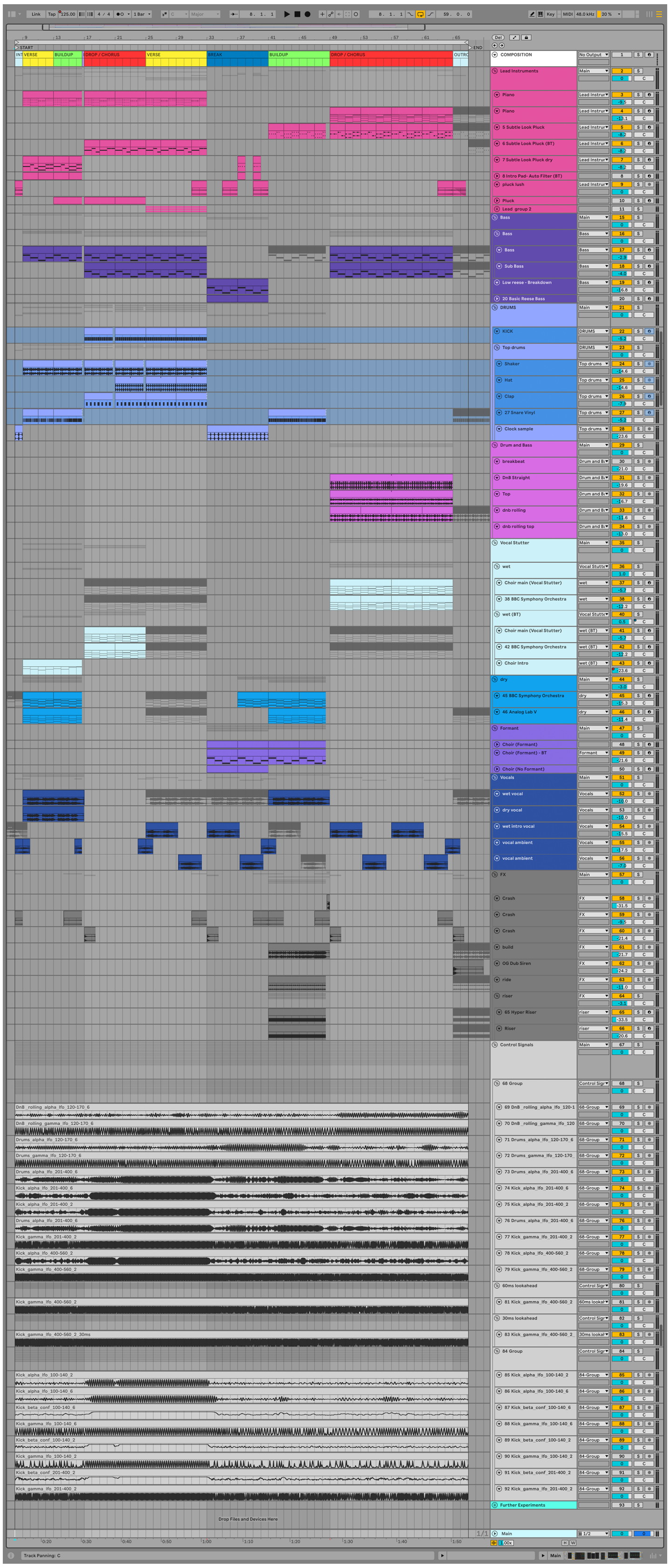


Figure D.1. Composition of multiple screenshots: demo project overview in Ableton Live Beta v.12.1b4.

Bibliography

[1] *Ableton live.* https://www.ableton.com/en/, Retrieved 11.06.2024.

[2] *Max 8 documentation.* https://docs.cycling74.com/max8, Retrieved 11.06.2024.

[3] *Mixed in key: Software for djs and music producers.* https://mixedinkey.com/, Retrieved 01.07.2024.

[4] *Ozone imager—free stereo imaging plug-in — izotope.* https://www.izotope.com/en/products/ozone-imager.html, Retrieved 28.07.2024.

[5] *Rekordbox 6.0.0 documentation.* https://cdn.rekordbox.com/files/20200323144552/rekordbox6.0.0_manual_EN.pdf, Retrieved 05.07.2024.

[6] *Slink devices — hypnus records.* https://hypnusrecords.com/shop, Retrieved 28.07.2024.

[7] *Vital audio - vital synthesizer plugin website.* https://vital.audio/, Retrieved 20.07.2024.

[8] J. D. R. Brecht De Man, Ryan Stables, *Intelligent Music Production*, Taylor Francis, Routlegde, 2020.

[9] P. Grosche and M. Müller, *Extracting predominant local pulse information from music recordings*, IEEE Transactions on Audio, Speech, and Language Processing, 19 (2011), pp. 1688–1701.

[10] R. Izhaki, *Mixing Audio: Concepts, Practices, and Tools*, Routledge, 4th ed., 2023.

[11] M. F. Matthew E. P. Davies, Sebastian Böck, *Audio metering : measurements, standards and practice* , Routledge, Taylor Francis Group, 3rd ed., 2020.

[12] A. Maz, *Music Technology Essentials: A Home Studio Guide* , Routledge, Taylor Francis Group, 2023.

[13] P. Meier, C.-Y. Chiu, and M. Müller, *A real-time beat tracking system with zero latency and enhanced controllability*, Transactions of the International Society for Music Information Retrieval (TISMIR), (In Review).

[14] P. Meier, G. Krump, and M. Müller, *A real-time beat tracking system based on predominant local pulse information*, in Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR), Online, 2021.

[15] P. Meier, S. Schwär, and M. Müller, *A real-time approach for estimating pulse tracking parameters for beat-synchronous audio effects*, in Proceedings of the International Conference on Digital Audio Effects (DAFx), Guildford, Surrey, UK, 2024.

[16] M. Müller, *Fundamentals of Music Processing - Using Python and Jupyter Notebooks, Second Edition*, Springer, 2021.

[17] M. Müller and F. Zalkow, *FMP notebooks: Educational material for teaching and learning fundamentals of music processing*, in Proceedings of the International Conference on Music Information Retrieval (ISMIR), Delft, The Netherlands, November 2019.

[18] A. Stark, M. Plumbley, and M. Davies, *Audio effects for real-time performance using beat tracking*, 05 2007.

[19] M. Steppat, *Audioprogrammierung Klangsynthese, Bearbeitung, Sounddesign*, https://www.hanser-elibrary.com/isbn/9783446432222, 2014.

[20] U. Zölzer, ed., *DAFX: Digital Audio Effects*, Wiley, 2nd ed., 2011.