CHAPTER 4

INTENSITY

Intensity, magnitude, and loudness-related features constitute one of the most commonly used classes for the description of audio content. Most audio editors and digital audio workstations illustrate the audio signal in its waveform view, its amplitude variation over time. There also exists a variety of instruments for level, volume, and loudness measurements frequently used in recording studio environments.

Many of the presented features are instantaneous features similar to the features introduced in Chap. 3. Therefore the same post-processing options (see Sect. 3.5) can be applied to most of the features introduced in the following.

4.1 Human Perception of Intensity and Loudness

It is important to distinguish the meaning of the terms *intensity* and *loudness*. Intensity means a physical, measurable entity such as the magnitude of a sound while loudness refers to a perceptual entity that can only be measured via responses of human observers [78]. Unfortunately, the term *loudness* is also used for *algorithmic models* of the perceived loudness.

Human perception of intensity is related to the magnitude of the audio signal in a way that if the signal's magnitude is scaled up, the perceived loudness will increase as well. It has been discovered very early that this relationship is non-linear; a linear increase in the signal's magnitude or power will not result in a linear increase in perceived loudness. An approximately linear relation could be found by using the pseudo-unit *decibel* (dB) which

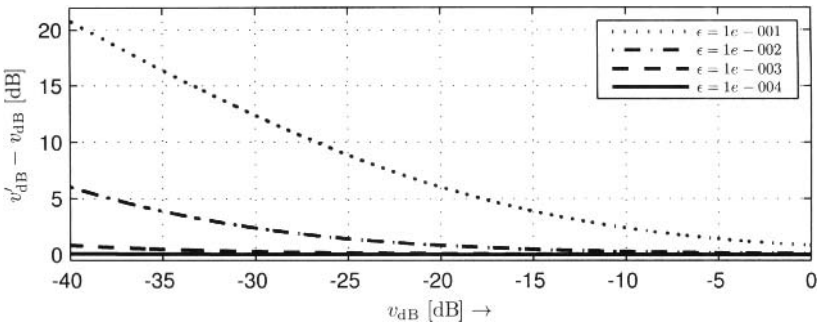

Figure 4.1 Level error introduced by adding a small constant ϵ to the argument of a logarithm

is computed by taking the logarithm of the intensity feature $v(n)$ (computed from a block of samples, see below):

$$v_{\mathrm{dB}}(n) = 20 \cdot \log_{10}\left(\frac{v(n)}{v_0}\right) \tag{4.1}$$

with v_0 representing a reference constant. In the digital domain, dealing with audio amplitudes in the range of $[-1; 1]$, it is commonly set to $v_0 = 1$. The resulting level unit is then referred to as dBFS (dB *full scale*) and has a range of $-\infty \leq v_{\mathrm{dB}}(n) \leq 0\,\mathrm{dB}$. The scaling factor 20 has been chosen to scale the non-linear function so that 1 dB roughly represents the level difference a human can easily recognize. This is only a rough approximation as the actual so-called *Just Noticeable Difference in Level (JNDL)* depends on the stimulus level and partly on stimulus frequency and masking effects [47].

Computing the logarithm of the feature v is not possible for silence $v(n) = 0$. The calculation of $\log(0)$ is commonly avoided by adding a small constant ϵ, resulting in

$$v_{\mathrm{dB}}(n) = 20 \cdot \log_{10}(v(n) + \epsilon). \tag{4.2}$$

The choice of ϵ determines the measurement accuracy at low-level inputs. The measurement error increases for decreasing $v(n)$ approaching ϵ. It will be 6 dB for $v(n) = \epsilon$ and increase with lower levels. Figure 4.1 visualizes the error amount for different ϵ. Alternatively, the input feature values may be truncated at ϵ to yield a correct value for magnitudes greater or equal than ϵ at the cost of an additional `if` statement per feature value:

$$v_{\mathrm{trunc}}(n) = \begin{cases} v(n), & \text{if } v(n) \geq \epsilon \\ \epsilon, & \text{otherwise} \end{cases}. \tag{4.3}$$

The decibel scale is not a loudness scale since equal-sized steps on the decibel scale are not perceived as equal-sized loudness steps by human listeners: doubling the level in dB does not result in doubling the perceived loudness of a sound. Stevens, summarizing several loudness perception studies, proposed a simple rule of thumb stating that a doubling of the perceived loudness corresponds to a level increase of 10 dB [79].

More accurate models of the human perception of loudness take both the input signal's frequency and the cochlea's frequency resolution into account as has been shown by many researchers during the last century. The most important researchers in the history of loudness perception are probably Fletcher and Munson [80, 81], Stevens [78, 79, 82], Moore [83], and Zwicker and Fastl [47].

4.2 Representation of Dynamics in Music

In a traditional musical score, loudness-related performance instructions are rather vague. Usually only five to eight different dynamic steps are used to describe musical dynamics (e.g., ***pp***: *pianissimo*, ***p***: *piano*, ***mf***: *mezzoforte*, ***f***: *forte*, ***ff***: *fortissimo* for the dynamic range from "very soft" to "very strong"), complemented by indications of smooth loudness transitions (e.g., *crescendo* or *decrescendo* for increasing and decreasing loudness, respectively) and dynamic accents (e.g., ***sf***: *sforzando*). These written instructions do not directly refer to absolute loudness as it would also depend on a number of other influencing factors such as instrumentation, timbre, number of voices, performance, and musical tension and musical context. The indifference to absolute loudness may also be illustrated by the fact that while listening to a recording on a hi-fi system, the reproduction volume may be manipulated without loosing the *piano* or *forte* character of the performance. Still, Nakamura has shown that measures of intensity or loudness can to a certain degree be used as indications of musical dynamics [84]. A loudness-related performance attribute is the *tremolo*, the periodic modulation of loudness over time. It usually appears in combination with a vibrato.

A more technical representation of dynamics in music is the *velocity* as standardized in the MIDI protocol [3]. It consists of 128 volume steps with the highest representing maximal intensity. But although the number of velocity steps is standardized, there is no standardized relationship between MIDI velocity and intensity: Goebl and Bresin found that for the Yamaha Disklavier and the Bösendorfer SE System (both pianos allowing for the monitoring of performance data), the relationship between MIDI velocity and (logarithmic) sound pressure level is nearly linear when disregarding very low and high values [85]. Dannenberg investigated the RMS peak level of various synthesizers and software instruments and found great differences among different synthesizers [86]. He identified a general trend for the velocity to be related to the square root of the RMS peak instead of its logarithm. Using one single electronic instrument, Taguti measured the A-weighted sound pressure level dependent on velocity and key [87]. The results, displayed over various keys for different input velocities, showed non-systematic deviations of up to 10 dB from a constant level among keys.

4.3 Features

The features presented in this section can be roughly structured into three categories: physical measures of sound intensity, approaches to measure intensity in recording studio environments, and psycho-acoustically motivated features modeling the human perception of loudness.

4.3.1 Root Mean Square

The *RMS* is one of the most common intensity features and is sometimes directly referred to as the sound intensity. It is calculated from a block of audio samples by

$$v_{\mathrm{RMS}}(n) = \sqrt{\frac{1}{\mathcal{K}} \sum_{i=i_s(n)}^{i_e(n)} x(i)^2}. \tag{4.4}$$

Table 4.1 RMS for the three prototypical signal types *silence* (zero magnitude at all samples), *white noise* with a rectangular PDF and a peak amplitude A, and a sinusoidal signal with the same peak amplitude

Input Signal	v_{RMS}
silence	0
rect. white noise (ampl. A)	$A/\sqrt{3}$
sinusoidal (ampl. A)	$A/\sqrt{2}$

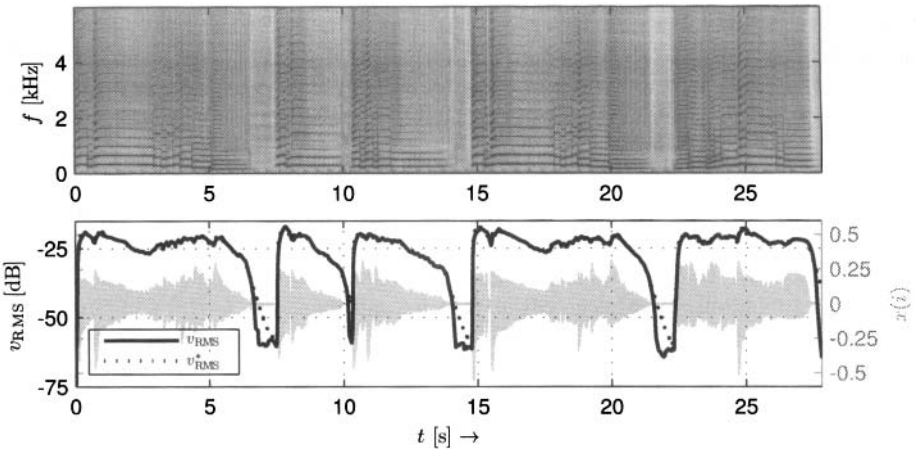

Figure 4.2 Spectrogram (top), waveform (bottom background), and RMS (bottom foreground) of a saxophone signal

Typical block lengths for the RMS calculation are in the range of several hundred milliseconds. The length in seconds is the so-called *integration time*.

The result of the calculation is a value within the range $0 \leq v_{\mathrm{RMS}}(n) \leq 1$ (as long as the amplitude 1 represents full scale. It will equal 0 if the input is silence and will approach 1 for both a square wave with maximum amplitude and a constant DC offset at ± 1. Sharp transients in the signal will be smoothed out by an RMS measure due to the comparably long integration time. Table 4.1 shows the RMS results for three signal prototypes.

Figure 4.2 shows the RMS of an example signal for two implementations, one calculated as shown above and the other with an approximation explained below. The RMS is a measure of the power of the signal; the two implementations shown are roughly equivalent except during sudden signal pauses in which the low-pass filtered variant only slowly decreases.

4.3.1.1 Common Variants

The calculation of the RMS can be computationally inefficient for large block lengths and small hop sizes $\mathcal{H}$. If the hop size is one sample, it is possible to reduce the number of computations by using the method of recursive implementation introduced in the context of the MA filter in Sect. 2.2.1.1:

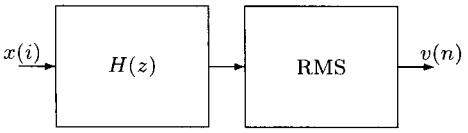

Figure 4.3 Flowchart of the frequency-weighted RMS calculation

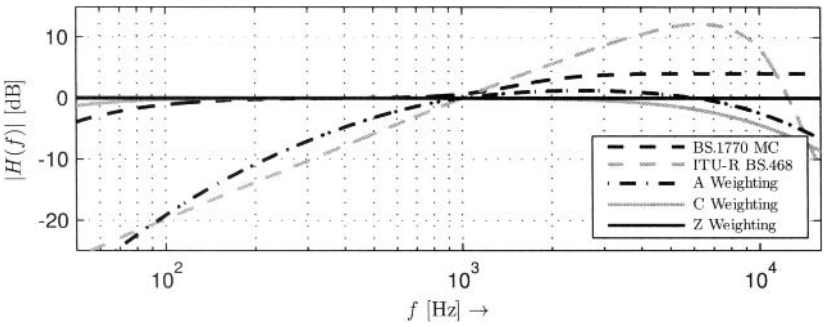

Figure 4.4 Frequency weighting transfer functions applied before RMS measurement

$$v^2_{\mathrm{RMS}}(n) = \frac{x(i_{\mathrm{e}}(n))^2 - x(i_{\mathrm{s}}(n-1))^2}{i_{\mathrm{e}}(n) - i_{\mathrm{s}}(n) + 1} + v^2_{\mathrm{RMS}}(n-1), \tag{4.5}$$

$$v_{\mathrm{RMS}}(n) = \sqrt{v^2_{\mathrm{RMS}}(n)}. \tag{4.6}$$

This implementation is computationally efficient but still requires a significant amount of memory to be allocated for large block lengths. An approximation of the RMS v^*_{RMS} with hop size $\mathcal{H} = 1$ can be implemented with a single-pole filter (compare Sect. 2.2.1.1):

$$v_{\mathrm{tmp}}(i) = \alpha \cdot v_{\mathrm{tmp}}(i-1) + (1-\alpha) \cdot x(i)^2 \tag{4.7}$$

$$v^*_{\mathrm{RMS}}(i) = \sqrt{v_{\mathrm{tmp}}(i)}. \tag{4.8}$$

The filter coefficient α can be estimated from the integration time (or block length) with Eq. (2.29).

The RMS computation is often preceded by a weighting filter. Figure 4.3 shows typical transfer functions for such weighting filters. The transfer function of a weighting filter is usually modeled after an inverse *equal-loudness contour* which measures the level for which a listener perceives equal loudness at different frequencies [80]. Thus, the weighting filter amplifies frequency regions in which the human ear is sensitive and attenuates other regions.

The specific weighting filter transfer functions shown in Fig. 4.4 are:

- *A weighting*: weighting function to be used for sounds at low level [88],
- *C weighting*: weighting function to be used for sounds at medium level [88],

- *Z weighting*: flat frequency weighting function (no weighting) [88],
- *RLB weighting*: weighting function according to ITU-R BS.1770 (plus high-frequency emphasis for multichannel signals) [89], and
- *CCIR weighting*: weighting function according to ITU-R BS.468 [90].

When frequency weighted RMS measures are used as models of loudness, the integration time is usually several seconds in order to ignore short-term variations of the signal.

4.3.2 Peak Envelope

The *peak envelope* of an audio signal can be extracted in different ways. The simplest way of extracting the envelope is to find the absolute maximum per block of audio samples:

$$v_{\text{Peak}}(n) = \max_{i_s(n) \le i \le i_e(n)} |x(i)|. \tag{4.9}$$

Using a so-called *Peak Program Meter (PPM)*, the envelope is extracted on a sample-per-sample basis. The PPM, frequently used in recording studio environments, operates with different integration times for attack (*attack time*) and release (*release time*). Typically, the attack time is significantly shorter than the release time (e.g., attack time: 10 ms, release time: 1500 ms) which means that the output reflects an increase in level faster than a decrease. This originates in requirements of recording engineers: the attack time has to be short in order to allow the systems to detect short peaks while the longer release time gives humans more time to actually see those peaks.

The structure of a digital PPM as described by Zölzer is shown in Fig. 4.5 [91]. The filter coefficient representing the attack time is named α_{AT}. The release time coefficient α_{RT} is not always in use and is being represented by λ in the figure. More specifically, λ has the following two states depending on the input magnitude

$$\lambda = \begin{cases} \alpha_{\text{RT}}, & \text{if } |x(i)| \le v_{\text{PPM}}(i-1) \\ 0, & \text{otherwise} \end{cases}. \tag{4.10}$$

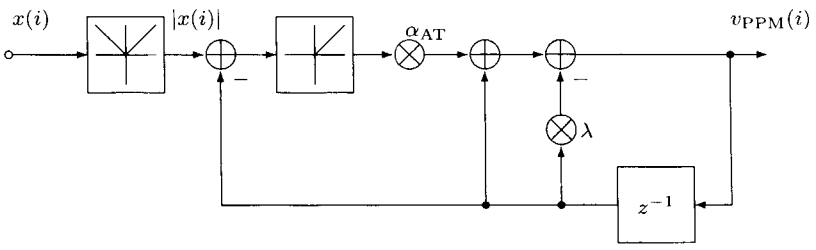

Figure 4.5 Flowchart of a Peak program meter

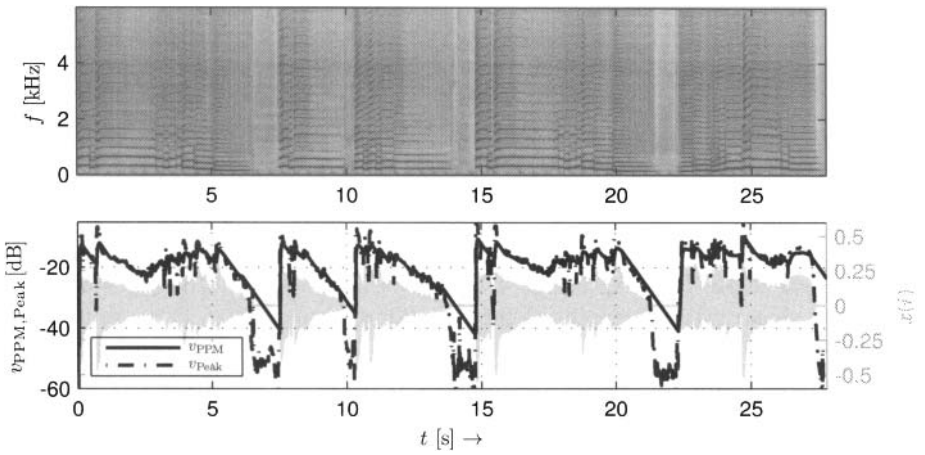

Figure 4.6 Spectrogram (top), waveform (bottom background), and PPM output compared to the magnitude maximum per block (bottom foreground) of a saxophone signal

The two states may be referred to as *release state* and *attack state*. The corresponding output equations are

- *Release state*:

$$\begin{aligned} v_{\mathrm{PPM}}(i) &= v_{\mathrm{PPM}}(i-1) - \alpha_{\mathrm{RT}} \cdot v_{\mathrm{PPM}}(i-1) \\ &= (1-\alpha_{\mathrm{RT}}) \cdot v_{\mathrm{PPM}}(i-1), \end{aligned} \tag{4.11}$$

- *Attack state*:

$$\begin{aligned} v_{\mathrm{PPM}}(i) &= \alpha_{\mathrm{AT}} \cdot \big(|x(i)| - v_{\mathrm{PPM}}(i-1)\big) + v_{\mathrm{PPM}}(i-1) \\ &= \alpha_{\mathrm{AT}} \cdot |x(i)| + (1-\alpha_{\mathrm{AT}}) \cdot v_{\mathrm{PPM}}(i-1). \end{aligned} \tag{4.12}$$

The result of both v_{Peak} and v_{PPM} is a value within the range $0 \leq v_{\mathrm{PPM}} \leq 1$. It will equal 0 if the input is silence and approach 1 for certain full-scale signals.

Figure 4.6 shows the results of both envelope measures. The comparison with the two RMS results as shown in Fig. 4.2 reveals similar behavior of RMS and peak measures as the calculation is relatively similar. The dotted peak maximum shows faster and more pronounced changes as can be clearly seen during the pauses with the PPM's constant decrease due to the long release time.

4.3.3 Psycho-Acoustic Loudness Features

There exist complex loudness measurements based on either psycho-acoustic properties of human loudness perception or physiological models of the human ear (or both). These are not frequently used for ACA, as (a) they are comparably costly to implement and to compute and (b) there are indications that they are in many cases equally meaningful as simpler loudness approximations such as the algorithm described in *International Telecommunication Union (ITU)* recommendation BS.1770, a weighted RMS solution [92].

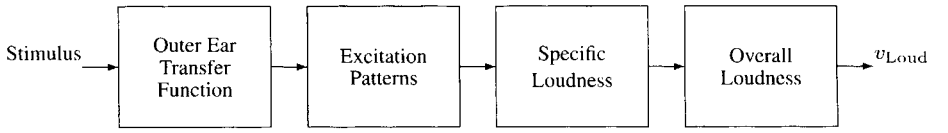

Figure 4.7 Flowchart of Zwicker's model for loudness computation

A widely known and psycho-acoustically motivated loudness measurement has been proposed by Zwicker [47, 93]. Figure 4.7 presents the flowchart of this loudness calculation. The signal is transformed into the bark domain (see Sect. 5.1) by a filterbank or a similar frequency transform. The so-called *excitation patterns* are computed from the filterbank outputs by taking into account frequency sensitivity and masking effects. Then, the *specific loudness* is calculated from the excitation patterns per band. The resulting loudness is the sum of the specific loudness over all bands.

4.3.3.1 EBU R128

One relatively recent recommendation for measurement of the loudness of a program or a file has been published by the *European Broadcasting Union (EBU)* [94]. The loudness itself is measured in compliance to ITU recommendation BS.1770, an RMS measure with an *RLB weighting*. The required block length is 3.0 s with a block overlap ratio of at least 66%. Blocks with very low loudness are discarded with a gating threshold.

The EBU recommendation also requires the following additional values to be extracted from the audio signal:

- clipped values according to an up-sampled true peak meter, and
- the loudness range computed from a histogram of all loudness block results as the difference $Q_v(0.95) - Q_v(0.10)$.