1

Introduction

V. Verfaille, M. Holters and U. Zölzer

1.1 Digital audio effects DAFX with MATLAB®

Audio effects are used by all individuals involved in the generation of musical signals and start with special playing techniques by musicians, merge to the use of special microphone techniques and migrate to effect processors for synthesizing, recording, production and broadcasting of musical signals. This book will cover several categories of sound or audio effects and their impact on sound modifications. Digital audio effects – as an acronym we use DAFX – are boxes or software tools with input audio signals or sounds which are modified according to some sound control parameters and deliver output signals or sounds (see Figure 1.1). The input and output signals are monitored by loudspeakers or headphones and some kind of visual representation of the signal, such as the time signal, the signal level and its spectrum. According to acoustical criteria the sound engineer or musician sets his control parameters for the sound effect he would like to achieve. Both input and output signals are in digital format and represent analog audio signals. Modification of the sound characteristic of the input signal is the main goal of digital audio effects. The settings of the control parameters are often done by sound engineers, musicians (performers, composers, or digital instrument makers) or simply the music listener, but can also be part of one specific level in the signal processing chain of the digital audio effect.

The aim of this book is the description of digital audio effects with regard to:

- *Physical and acoustical effect:* we take a short look at the physical background and explanation. We describe analog means or devices which generate the sound effect.
- *Digital signal processing:* we give a formal description of the underlying algorithm and show some implementation examples.
- *Musical applications:* we point out some applications and give references to sound examples available on CD or on the web.

DAFX: Digital Audio Effects, Second Edition. Edited by Udo Zölzer.

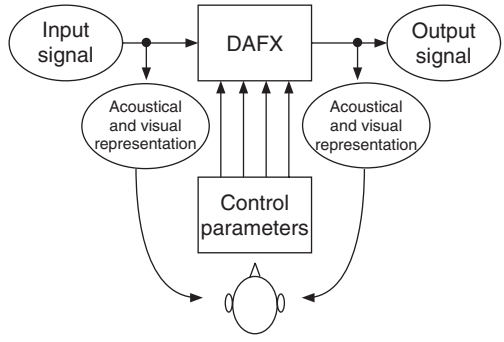

Figure 1.1 Digital audio effect and its control [Arf99].

The physical and acoustical phenomena of digital audio effects will be presented at the beginning of each effect description, followed by an explanation of the signal processing techniques to achieve the effect, some musical applications and the control of effect parameters.

In this introductory chapter we next introduce some vocabulary clarifications, and then present an overview of classifications of digital audio effects. We then explain some simple basics of digital signal processing and show how to write simulation software for audio effects processing with the **MATLAB**[1] simulation tool or freeware simulation tools[2]. **MATLAB** implementations of digital audio effects are a long way from running in real time on a personal computer or allowing real-time control of its parameters. Nevertheless the programming of signal processing algorithms and in particular sound-effect algorithms with **MATLAB** is very easy and can be learned very quickly.

Sound effect, audio effect and sound transformation

As soon as the word "effect" is used, the viewpoint that stands behind is the one of the subject who is observing a phenomenon. Indeed, "effect" denotes an impression produced in the mind of a person, a change in perception resulting from a cause. Two uses of this word denote related, but slightly different aspects: "sound effects" and "audio effects." Note that in this book, we discuss the latter exclusively. The expression – "sound effects" – is often used to depict sorts of earcones (icons for the ear), special sounds which in production mode have a strong signature and which therefore are very easily identifiable. Databases of sound effects provide natural (recorded) and processed sounds (resulting from sound synthesis and from audio effects) that produce specific effects on perception used to simulate actions, interaction or emotions in various contexts. They are, for instance, used for movie soundtracks, for cartoons and for music pieces. On the other hand, the expression "audio effects" corresponds to the tool that is used to apply transformations to sounds in order to modify how they affect us. We can understand those two meanings as a shift of the meaning of "effect": from the perception of a change itself to the signal processing technique that is used to achieve this change of perception. This shift reflects a semantic confusion between the object (what is perceived) and the tool to make the object (the signal processing technique). "Sound effect" really deals with the subjective viewpoint, whereas "audio effect" uses a subject-related term (effect) to talk about an objective reality: the tool to produce the sound transformation.

Historically, it can arguably be said that audio effects appeared first, and sound transformations later, when this expression was tagged on refined sound models. Indeed, techniques that made use of an analysis/transformation/synthesis scheme embedded a transformation step performed on a refined model of the sound. This is the technical aspect that clearly distinguishes "audio effects"

[1] http://www.mathworks.com
[2] http://www.octave.org

and "sound transformations," the former using a simple representation of the sound (samples) to perform signal processing, whereas the latter uses complex techniques to perform enhanced signal processing. Audio effects originally denoted simple processing systems based on simple operations, e.g. chorus by random control of delay line modulation; echo by a delay line; distortion by non-linear processing. It was assumed that audio effects process sound at its *surface*, since sound is represented by the wave form samples (which is not a high-level sound model) and simply processed by delay lines, filters, gains, etc. By surface we do not mean how strongly the sound is modified (it in fact can be deeply modified; just think of distortion), but we mean how far we go in unfolding the sound representations to be accurate and refined in the data and model parameters we manipulate. Sound transformations, on the other hand, denoted complex processing systems based on analysis/transformation/synthesis models. We, for instance, think of the phase vocoder with fundamental frequency tracking, the source-filter model, or the sinusoidal plus residual additive model. They were considered to offer deeper modifications, such as high-quality pitch-shifting with formant preservation, timbre morphing, and time-scaling with attack, pitch and panning preservation. Such deep manipulation of control parameters allows in turn the sound modifications to be heard as very subtle.

Over time, however, practice blurred the boundaries between audio effects and sound transformations. Indeed, several analysis/transformation/synthesis schemes can simply perform various processing that we consider to be audio effects. On the other hand, usual audio effects such as filters have undergone tremendous development in terms of design, in order to achieve the ability to control the frequency range and the amplitude gain, while taking care to limit the phase modulation. Also, some usual audio effects considered as simple processing actually require complex processing. For instance, reverberation systems are usually considered as simple audio effects because they were originally developed using simple operations with delay lines, even though they apply complex sound transformations. For all those reasons, one may consider that the terms "audio effects," "sound transformations" and "musical sound processing" are all refering to the same idea, which is to apply signal processing techniques to sounds in order to modify how they will be perceived, or in other words, to transform a sound into another sound with a perceptually different quality. While the different terms are often used interchangeably, we use "audio effects" throughout the book for the sake of consistency.

1.2 Classifications of DAFX

Digital audio effects are mainly used by composers, performers and sound engineers, but they are generally described from the standpoint of the DSP engineers who designed them. Therefore, their classification and documentation, both in software documentation and textbooks, rely on the underlying techniques and technologies. If we observe what happens in different communities, there exist other classification schemes that are commonly used. These include signal processing classification [Orf96, PPPR96, Roa96, Moo90, Zöl02], control type classification [VWD06], perceptual classification [ABL+03], and sound and music computing classification [CPR95], among others. Taking a closer look in order to compare these classifications, we observe strong differences. The reason is that each classification has been introduced in order to best meet the needs of a specific audience; it then relies on a series of features. Logically, such features are relevant for a given community, but may be meaningless or obscure for a different community. For instance, signal-processing techniques are rarely presented according to the perceptual features that are modified, but rather according to acoustical dimensions. Conversely, composers usually rely on perceptual or cognitive features rather than acoustical dimensions, and even less on signal-processing aspects.

An interdisciplinary approach to audio effect classification [VGT06] aims at facilitating the communication between researchers and creators that are working on or with audio effects.[3] Various

[3] e.g. DSP programmers, sound engineers, sound designers, electroacoustic music composers, performers using augmented or extended acoustic instruments or digital instruments, musicologists.

disciplines are then concerned: from acoustics and electrical engineering to psychoacoustics, music cognition and psycholinguistics. The next subsections present the various standpoints on digital audio effects through a description of the communication chain in music. From this viewpoint, three discipline-specific classifications are described: based on underlying techniques, control signals and perceptual attributes, then allowing the introduction of interdisciplinary classifications linking the different layers of domain-specific descriptors. It should be pointed out that the presented classifications are not classifications *stricto sensu*, since they are neither exhaustive nor mutually exclusive: one effect can be belong to more than one class, depending on other parameters such as the control type, the artefacts produced, the techniques used, etc.

Communication chain in music

Despite the variety of needs and standpoints, the technological terminology is predominantly employed by the actual users of audio effects: composers and performers. This technological classification might be the most rigorous and systematic one, but it unfortunately only refers to the techniques used, while ignoring our perception of the resulting audio effects, which seems more relevant in a musical context.

We consider the communication chain in music that essentially produces musical sounds [Rab, HMM04]. Such an application of the communication-chain concept to music has been adapted from linguistics and semiology [Nat75], based on Molino's work [Mol75]. This adaptation in a tripartite semiological scheme distinguishes three levels of musical communication between a composer (producer) and a listener (receiver) through a physical, neutral trace such as a sound. As depicted in Figure 1.2, we apply this scheme to a complete chain in order to investigate all possible standpoints on audio effects. In doing so, we include all actors intervening in the various processes of the conception, creation and perception of music, who are instrument-makers, composers, performers and listeners. The *poietic level* concerns the conception and creation of a *musical message* to which instrument-makers, composers and performers participate in different ways and at different stages. The *neutral level* is that of the physical "trace" (instruments, sounds or scores). The *aesthetic level* corresponds to the perception and reception of the *musical message* by a listener. In the case of audio effects, the instrument-maker is the signal-processing engineer who designs the effect and the performer is the user of the effect (musician, sound engineer). In the context of home studios and specific musical genres (such as mixed music creation), composers, performers and instrument-makers (music technologists) are usually distinct individuals who need to efficiently communicate with one another. But all actors in the chain are also listeners who can share descriptions of what they hear and how they interpret it. Therefore we will consider the perceptual and cognitive standpoints as the entrance point to the proposed interdisciplinary network of the various domain-specific classifications. We also consider the specific case of the home studio where a performer may also be his very own sound engineer, designs or sets his processing chain, and performs the mastering. Similarly, electroacoustic music composers often combine such tasks with additional programming and performance skills. They conceive their own processing system, control and perform on their instruments. Although all production tasks are performed by a single multidisciplinary artist in these two cases, a transverse classification is still helpful to achieve a

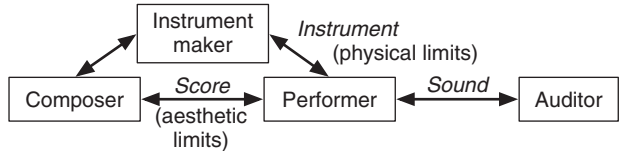

Figure 1.2 Communication chain in music: the composer, performer and instrument maker are also listeners, but in a different context than the auditor.

better awareness of the relations, between the different description levels of an audio effect, from technical to perceptual standpoints.

1.2.1 Classification based on underlying techniques

Using the standpoint of the "instrument-maker" (DSP engineer or software engineer), this first classification focuses on the underlying techniques that are used in order to implement the audio effects. Many digital implementations of audio effects are in fact emulations of their analog ancestors. Similarly, some analog audio effects implemented with one technique were emulating audio effects that already existed with another analog technique. Of course, at some point analog and/or digital techniques were also creatively used so as to provide new effects. We can distinguish the following analog technologies, in chronological order:

- Mechanics/acoustics (e.g., musical instruments and effects due to room acoustics)
- Electromechanics (e.g., using vinyls)
- Electromagnetics (e.g., flanging and time-scaling with magnetic tapes)
- Electronics (e.g., filters, vocoder, ring modulators).

With mechanical means, such as designing or choosing a specific room for its acoustical properties, music was modified and shaped to the wills of composers and performers. With electromechanical means, vinyls could be used to time-scale and pitch-shift a sound by changing disk rotation speed.[4] With electromagnetic means, flanging was originally obtained when pressing the thumb on the flange of a magnetophone wheel[5] and is now emulated with digital comb filters with varying delays. Another example of electromagnetic means is the time-scaling effect without pitch-shifting (i.e., with "not-too-bad" timbre preservation) performed by the composer and engineer Pierre Schaeffer back in the early 1950s. Electronic means include ring modulation, which refers to the multiplication of two signals and borrows its name from the analog ring-shaped circuit of diodes originally used to implement this effect.

Digital effects emulating acoustical or perceptual properties of electromechanic, electric or electronic effects include filtering, the wah-wah effect,[6] the vocoder effect, reverberation, echo and the Leslie effect. More recently, electronic and digital sound processing and synthesis allowed for the creation of new unprecedented effects, such as robotization, spectral panoramization, prosody change by adaptive time-scaling and pitch-shifting, and so on. Of course, the boundaries between imitation and creative use of technology is not clear cut. The vocoding effect, for example, was first developed to encode voice by controlling the spectral envelope with a filter bank, but was later used for musical purposes, specifically to add a vocalic aspect to a musical sound. A digital synthesis counterpart results from a creative use (LPC, phase vocoder) of a system allowing for the imitation of acoustical properties. Digital audio effects can be organized on the basis of implementation techniques, as it is proposed in this book:

- Filters and delays (resampling)
- Modulators and demodulators

[4] Such practice was usual in the first cinemas with sound, where the person in charge of the projection was synchronizing the sound to the image, as explained with a lot of humor by the awarded filmmaker Peter Brook in his autobiography: Threads of Time: Recollections, 1998.

[5] It is considered that flanging was first performed by George Martin and the Beatles, when John Lennon was asking for a technical way to replace dubbing.

[6] It seems that the term wah-wah was first coined by Miles Davis in the 1950s to describe how he manipulated sound with his trumpet's mute.

- Non-linear processing
- Spatial effects
- Time-segment processing
- Time-frequency processing
- Source-filter processing
- Adaptive effects processing
- Spectral processing
- Time and frequency warping
- Virtual analog effects
- Automatic mixing
- Source separation.

Another classification of digital audio effects is based on the domain where the signal processing is applied (namely time, frequency and time-frequency), together with the indication whether the processing is performed sample-by-sample or block-by-block:

- Time domain:
  - block processing using overlap-add (OLA) techniques (e.g., basic OLA, synchronized OLA, pitch synchronized OLA)
  - sample processing (filters, using delay lines, gain, non-linear processing, resampling and interpolation)
- Frequency domain (with block processing):
  - frequency-domain synthesis with inverse Fourier transform (e.g., phase vocoder with or without phase unwrapping)
  - time-domain synthesis (using oscillator bank)
- Time and frequency domain (e.g., phase vocoder plus LPC).

The advantage of such kinds of classification based on the underlying techniques is that the software developer can easily see the technical and implementation similarities of various effects, thus simplifying both the understanding and the implementation of multi-effect systems, which is depicted in the diagram in Figure 1.3. It also provides a good overview of technical domains and signal-processing techniques involved in effects. However, several audio effects appear in two places in the diagram (illustrating once again how these diagrams are not real classifications), belonging to more than a single class, because they can be performed with techniques from various domains. For instance, time-scaling can be performed with time-segment processing as well as with time-frequency processing. One step further, adaptive time-scaling with time-synchronization [VZA06] can be performed with SOLA using either block-by-block or time-domain processing, but also with the phase vocoder using a block-by-block frequency-domain analysis with IFFT synthesis.

Depending on the user expertise (DSP programmer, electroacoustic composer), this classification may not be the easiest to understand, even more since this type of classification does not explicitly handle perceptual features, which are the common vocabulary of all listeners. Another reason for introducing the perceptual attributes of sound in a classification is that when users can choose between various implementations of an effect, they also make their choice depending on

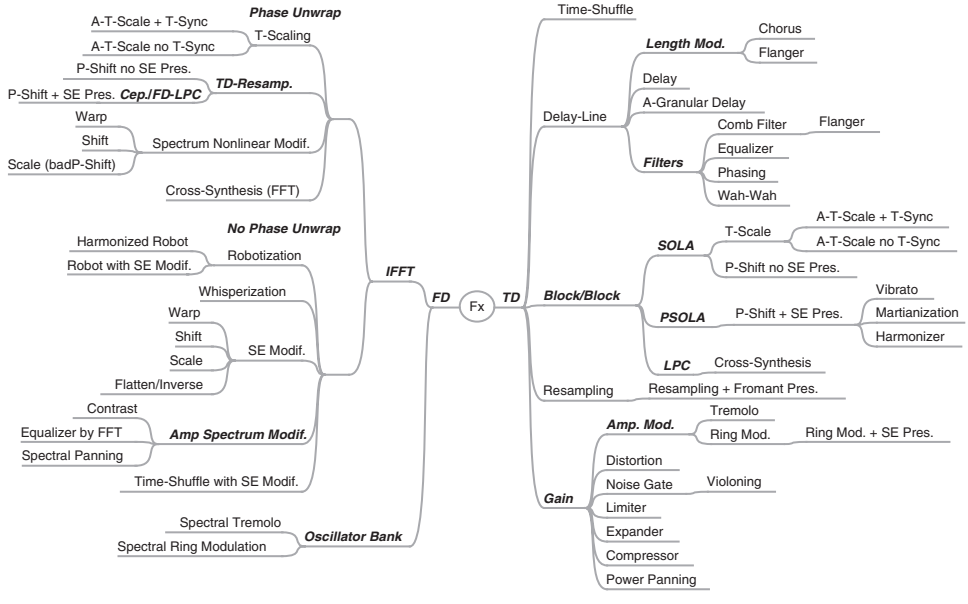

Figure 1.3 A technical classification of audio effects that could be used to design multi-effect systems. "TD" stands for "time-domain," "FD" for "frequency domain," "t-scale" for "time-scaling," "p-shift" for "pitch-shifting," "+" for "with," "A-" for adaptive control,'' "SE" for "spectral envelope," "osc" for "oscillator," "mod." for "modulation" and "modif." for "modification." Bold italic font words denote technical aspects, whereas regular font words denote audio effects.

the audible artifacts of each effect. For instance, with time-scaling, resampling does not preserve pitch nor formants; OLA with circular buffer adds the window modulation and sounds rougher and filtered; a phase vocoder sounds a bit reverberant, the "sinusoidal + noise" additive model sounds good except for attacks, the "sinusoidal + transients + noise" additive model preserves attacks, but not the spatial image of multi-channel sounds, etc. Therefore, in order to choose a technique, the user must be aware of the audible artifact of each technique. The need to link implementation techniques to perceptual features thus becomes clear and will be discussed next.

1.2.2 Classification based on perceptual attributes

Using the perceptual categorization, audio effects can be classified according to the perceptual attribute that is mainly altered by the digital processing (examples of "musical gestures" are also provided):

- Loudness: related to dynamics, nuances and phrasing (*legato*, and *pizzicato*), accents, *tremolo*
- Time: related to duration, tempo, and rhythmic modifications (*accelerando*, *deccelerando*)
- Pitch: composed of height and chroma, related to and organized into melody, intonation and harmony; sometimes shaped with *glissandi*
- Spatial hearing: related to source localization (distance, azimuth, elevation), motion (Doppler) and directivity, as well as to the room effect (reverberation, echo)

- Timbre: composed of short-term time features (such as transients and attacks) and long-term time features that are formants (color) and ray spectrum properties (texture, harmonicity), both coding aspects such as brightness (or spectral height), sound quality, timbral metamorphosis; related musical gestures contain various playing modes, ornamentation and special effects such as *vibrato*, trill, flutter tonguing, *legato, pizzicato*, harmonic notes, multiphonics, etc.

We consider this classification to be among the most natural to musicians and audio listeners, since such perceptual attributes are usually clearly identified in music scores. It has already been used to classify content-based transformations [ABL$^+$03] as well as adaptive audio effects [VZA06]. Therefore, we now discuss a more detailed overview of those perceptual attributes by highlighting some basics of psychoacoustics for each perceptual attribute. We also name commonly used digital audio effects, with a specific emphasis on timbre, as this more complex perceptive attribute offers the widest range of sound possibilities. We also highlight the relationships between perceptual attributes (or high-level features) and their physical counterparts (signal or low-level features), which are usually simpler to compute.

Loudness: Loudness is the perceived intensity of the sound through time. Its computational models perform time and frequency integration of the energy in critical bands [ZS65, Zwi77]. The sound intensity level computed by RMS (root mean square) is its physical counterpart. Using an additive analysis and a transient detection, we extract the sound intensity levels of the harmonic content, the transient and the residual. We generally use a logarithmic scale named decibels: loudness is then $L_{dB} = 20 \log_{10} I$, with I the intensity. Adding 20 dB to the loudness is obtained by multiplying the sound intensity level by 10. The musical counterpart of loudness is called dynamics, and corresponds to a scale ranging from *pianissimo* (*pp*) to *fortissimo* (*ff*) with a 3 dB space between two successive dynamic levels. *Tremolo* describes a loudness modulation with a specific frequency and depth. Commonly used loudness effects modify the sound intensity level: the volume change, the tremolo, the compressor, the expander, the noise gate and the limiter. The tremolo is a sinusoidal amplitude modulation of the sound intensity level with a modulation frequency between 4 and 7 Hz (around the 5.5 Hz frequency modulation of the vibrato). The compressor and the expander modify the intensity level using a non-linear function; they are among the first adaptive effects that were created. The former compresses the intensity level, thus giving more percussive sounds, whereas the latter has the opposite effect and is used to extend the dynamic range of the sound. With specific non-linear functions, we obtain noise gate and limiter effects. The noise gate bypasses sounds with very low loudness, which is especially useful to avoid the background noise that circulate throughout an effect system involving delays. Limiting the intensity level protects the hardware. Other forms of loudness effects include automatic mixers and automatic volume/gain control, which are sometimes noise-sensor equipped.

Time and Rhythm: Time is perceived through two intimately intricate attributes: the duration of sound and gaps, and the rhythm, which is based on repetition and inference of patterns [DH92]. Beat can be extracted with autocorrelation techniques, and patterns with quantification techniques [Lar01]. Time-scaling is used to fit the signal duration to a given duration, thus affecting rhythm. Resampling can perform time-scaling, resulting in an unwanted pitch-shifting. The time-scaling ratio is usually constant, and greater than 1 for time-expanding (or time-stretching, time-dilatation: sound is slowed down) and lower than 1 for time-compressing (or time-contraction: sound is sped up). Three block-by-block techniques avoid this: the phase vocoder [Por76, Dol86, AKZ02a], SOLA [MC90, Lar98] and the additive model [MQ86, SS90, VLM97]. Time-scaling with the phase vocoder technique consists of using different analysis and synthesis step increments. The phase vocoder is performed using the short-time Fourier transform (STFT) [AR77]. In the analysis step, the STFT of windowed input blocks is performed with an R_A samples step increment. In the synthesis step, the inverse Fourier transform delivers output blocks which are windowed, overlapped and then added with an R_S samples step increment. The phase vocoder step increments have to be suitably chosen to provide a perfect reconstruction of the signal [All77, AR77]. Phase

computation is needed for each frequency bin of the synthesis STFT. The phase vocoder technique can time-scale any type of sound, but adds phasiness if no care is taken: a peak phase-locking technique solves this problem [Puc95, LD97]. Time-scaling with the SOLA technique[7] is performed by duplication or suppression of temporal grains or blocks, with pitch synchronization of the overlapped grains in order to avoid low frequency modulation due to phase cancellation. Pitch-synchronization implies that the SOLA technique only correctly processes the monophonic sounds. Time-scaling with the additive model results in scaling the time axis of the partial frequencies and their amplitudes. The additive model can process harmonic as well as inharmonic sounds while having a good quality spectral line analysis.

Pitch: Harmonic sounds have their pitch given by the frequencies and amplitudes of the harmonics; the fundamental frequency is the physical counterpart. The attributes of pitch are height (high/low frequency) and chroma (or color) [She82]. A musical sound can be either perfectly harmonic (e.g., wind instruments), nearly harmonic (e.g., string instruments) or inharmonic (e.g., percussions, bells). Harmonicity is also related to timbre. Psychoacoustic models of the perceived pitch use both the spectral information (frequency) and the periodicity information (time) of the sound [dC04]. The pitch is perceived in the quasi-logarithmic *mel* scale, which is approximated by the log-Hertz scale. Tempered scale notes are transposed up by one octave when multiplying the fundamental frequency by 2 (same chroma, doubling the height). The pitch organization through time is called melody for monophonic sounds and harmony for polyphonic sounds. The pitch of harmonic sounds can be shifted, thus transposing the note. Pitch-shifting is the dual transformation of time-scaling, and consists of scaling the frequency axis of a time-frequency representation of the sound. A pitch-shifting ratio greater than 1 transposes up; lower than 1 it transposes down. It can be performed by a combination of time-scaling and resampling. In order to preserve the timbre and the spectral envelope [AKZ02b], the phase vocoder decomposes the signal into source and filter for each analysis block: the formants are pre-corrected (in the frequency domain [AD98]), the source signal is resampled (in the time domain) and phases are wrapped between two successive blocks (in the frequency domain). The PSOLA technique preserves the spectral envelope [BJ95, ML95], and performs pitch-shifting by using a synthesis step increment that differs from the analysis step increment. The additive model scales the spectrum by multiplying the frequency of each partial by the pitch-shifting ratio. Amplitudes are then linearly interpolated from the spectral envelope. Pitch-shifting of inharmonic sounds such as bells can also be performed by ring modulation. Using a pitch-shifting effect, one can derive harmonizer and auto tuning effects. Harmonizing consists of mixing a sound with several pitch-shifted versions of it, to obtain chords. When controlled by the input pitch and the melodic context, it is called smart harmony [AOPW99] or intelligent harmonization.[8] Auto tuning[9] consists of pitch-shifting a monophonic signal so that the pitch fits to the tempered scale [ABL+03].

Spatial Hearing: Spatial hearing has three attributes: the location, the directivity, and the room effect. The sound is localized by human beings with regards to distance, elevation and azimuth, through interaural intensity (IID) and inter-aural time (ITD) differences [Bla83], as well as through filtering via the head, the shoulders and the rest of the body (head-related transfer function, HRTF). When moving, sound is modified according to pitch, loudness and timbre, indicating the speed and direction of its motion (Doppler effect) [Cho71]. The directivity of a source is responsible for the differences in transfer functions according to the listener position relative to the source. The sound is transmitted through a medium as well as reflected, attenuated and filtered by obstacles (reverberation and echoes), thus providing cues for deducing the geometrical and material properties of the room. Spatial effects describe the spatialization of a sound with headphones or loudspeakers. The position in the space is simulated using intensity panning (e.g., constant power panoramization

[7] When talking about SOLA techniques, we refer to all the synchronized and overlap-add techniques: SOLA, TD-PSOLA, TF-PSOLA, WSOLA, etc.

[8] http://www.tc-helicon.tc/

[9] http://www.antarestech.com/

with two loudspeakers or headphones [Bla83], vector-based amplitude panning (VBAP) [Pul97] or Ambisonics [Ger85] with more loudspeakers), delay lines to simulate the precedence effect due to ITD, as well as filters in a transaural or binaural context [Bla83]. The Doppler effect is due to the behaviour of sound waves approaching or going away; the sound motion throughout the space is simulated using amplitude modulation, pitch-shifting and filtering [Cho71, SSAB02]. Echoes are created using delay lines that can eventually be fractional [LVKL96]. The room effect is simulated with artificial reverberation units that use either delay-line networks or all-pass filters [SL61, Moo79] or convolution with an impulse response. The simulation of instruments' directivity is performed with linear combination of simple directivity patterns of loudspeakers [WM01]. The rotating speaker used in the Leslie/Rotary is a directivity effect simulated as a Doppler [SSAB02].

Timbre: This attribute is difficult to define from a scientific point of view. It has been viewed for a long time as "that attribute of auditory sensation in terms of which a listener can judge that two sounds similarly presented and having the same loudness and pitch are dissimilar" [ANS60]. However, this does not take into account some basic facts, such as the ability to recognize and to name any instrument when hearing just one note or listening to it through a telephone [RW99]. The frequency composition of the sound is concerned, with the attack shape, the steady part and the decay of a sound, the variations of its spectral envelope through time (e.g., variations of formants of the voice), and the phase relationships between harmonics. These phase relationships are responsible for the whispered aspect of a voice, the roughness of low-frequency modulated signals, and also for the phasiness[10] introduced when harmonics are not phase aligned. We consider that timbre has several other attributes, including:

- The brightness or spectrum height, correlated to spectral centroid[11] [MWdSK95], and computed with various models [Cab99]
- The quality and noisiness, correlated to the signal-to-noise ratio (e.g., computed as the ratio between the harmonics and the residual intensity levels [ABL$^+$03]) and to the voiciness (computed from the autocorrelation function [BP89] as the second-highest peak value of the normalized autocorrelation)
- The texture, related to jitter and shimmer of partials/harmonics [DT96] (resulting from a statistical analysis of the partials' frequencies and amplitudes), to the balance of odd/even harmonics (given as the peak of the normalized autocorrelation sequence situated half way between the first- and second-highest peak values [AKZ02b]) and to harmonicity
- The formants (especially vowels for the voice [Sun87]) extracted from the spectral envelope, the spectral envelope of the residual and the mel-frequency critical bands (MFCC), perceptual correlate of the spectral envelope.

Timbre can be verbalized in terms of roughness, harmonicity, as well as openness, acuteness and laxness for the voice [Sla85]. At a higher level of perception, it can also be defined by musical aspects such as *vibrato* [RDS$^+$99], *trill* and *Flatterzunge*, and by note articulation such as *appoyando, tirando* and *pizzicato*.

Timbre effects is the widest category of audio effects and includes vibrato, chorus, flanging, phasing, equalization, spectral envelope modifications, spectral warping, whisperization, adaptive filtering and transient enhancement or attenuation.

- Vibrato is used for emphasis and timbral variety [MB90], and is defined as a complex timbre pulsation or modulation [Sea36] implying frequency modulation, amplitude modulation and

[10] Phasiness is usually involved in speakers reproduction, where phase inproperties make the sound poorly spatialized. In the phase vocoder technique, the phasiness refers to a reverberation artifact that appears when neighboring frequency bins representing the same sinusoid have different phase unwrapping.

[11] The spectral centroid is also correlated to other low-level features: the spectral slope, the zero-crossing rate, the high frequency content [MB96].

sometimes spectral-shape modulation [MB90, VGD05], with a nearly sinusoidal control. Its modulation frequency is around 5.5 Hz for the singing voice [Hon95]. Depending on the instruments, the vibrato is considered as a frequency modulation with a constant spectral shape (e.g., voice, [Sun87], stringed instruments [MK73, RW99]), an amplitude modulation (e.g., wind instruments), or a combination of both, on top of which may be added a complex spectral-shape modulation, with high-frequency harmonics enrichment due to non-linear properties of the resonant tube (voice [MB90], wind and brass instruments [RW99]).

- A chorus effect appears when several performers play together the same piece of music (same in melody, rhythm, dynamics) with the same kind of instrument. Slight pitch, dynamic, rhythm and timbre differences arise because the instruments are not physically identical, nor are perfectly tuned and synchronized. It is simulated by adding to the signal the output of a randomly modulated delay line [Orf96, Dat97]. A sinusoidal modulation of the delay line creates a flanging or sweeping comb filter effect [Bar70, Har78, Smi84, Dat97]. Chorus and flanging are specific cases of phase modifications known as phase shifting or phasing.

- Equalization is a well-known effect that exists in most of the sound systems. It consists in modifying the spectral envelope by filtering with the gains of a constant-Q filter bank. Shifting, scaling or warping of the spectral envelope is often used for voice sounds since it changes the formant places, yielding to the so-called Donald Duck effect [AKZ02b].

- Spectral warping consists of modifying the spectrum in a non-linear way [Fav01], and can be achieved using the additive model or the phase vocoder technique with peak phase-locking [Puc95, LD97]. Spectral warping allows for pitch-shifting (or spectrum scaling), spectrum shifting, and in-harmonizing.

- Whisperization transforms a spoken or sung voice into a whispered voice by randomizing either the magnitude spectrum or the phase spectrum of a short-time Fourier transform [AKZ02a]. Hoarseness is a quite similar effect that takes advantage of the additive model to modify the harmonic-to-residual ratio [ABL+03].

- Adaptive filtering is used in telecommunications [Hay96] in order to avoid the feedback loop effect created when the output signal of the telephone loudspeaker goes into the microphone. Filters can be applied in the time domain (comb filters, vocal-like filters, equalizer) or in the frequency domain (spectral envelope modification, equalizer).

- Transient enhancement or attenuation is obtained by changing the prominence of the transient compared to the steady part of a sound, for example using an enhanced compressor combined with a transient detector.

Multi-Dimensional Effects: Many other effects modify several perceptual attributes of sounds simultaneously. For example, robotization consists of replacing a human voice with a metallic machine-like voice by adding roughness, changing the pitch and locally preserving the formants. This is done using the phase vocoder and zeroing the phase of the grain STFT with a step increment given as the inverse of the fundamental frequency. All the samples between two successive non overlapping grains are zeroed[12] [AKZ02a]. Resampling consists of interpolating the wave form, thus modifying duration, pitch and timbre (formants). Ring modulation is an amplitude modulation without the original signal. As a consequence, it duplicates and shifts the spectrum and modifies pitch and timbre, depending on the relationship between the modulation frequency and the signal fundamental frequency [Dut91]. Pitch-shifting without preserving the spectral envelope modifies

[12] The robotization processing preserves the spectral shape of a processed grain at the local level. However, the formants are slightly modified at the global level because of overlap-add of grains with non-phase-aligned grain (phase cancellation) or with zeros (flattening of the spectral envelope).

both pitch and timbre. The use of multi-tap monophonic or stereophonic echoes allow for rhythmic, melodic and harmonic constructions through superposition of delayed sounds.

Summary of Effects by Perceptual Attribute: For the main audio effects, Tables 1.1, 1.2, and 1.3 indicate the perceptual attributes modified, along with complementary information for programmers and users about real-time implementation and control type. When the user chooses an effect to modify one perceptual attribute, the implementation technique used may introduce artifacts, implying modifications of other attributes. For that reason, we differentiate the perceptual attributes that we primarily want to modify ("main" perceptual attributes, and the corresponding dominant modification perceived) and the "secondary" perceptual attributes that are slightly modified (on purpose or as a by-product of the signal processing).

Table 1.1 Digital audio effects according to modified perceptual attributes (L for loudness, D for duration and rhythm, P for pitch and harmony, T for timbre and quality, and S for spatial qualities). We also indicate if real-time implementation (RT) is not possible (using "—"), and the built-in control type (A for adaptive, cross-A for cross-adaptive, and LFO for low-frequency oscillator).

Effect name	Perceptual Attributes		RT	Control
	Main	Other		
Effects mainly on loudness (L)				
compressor, limiter, expander, noise gate	L	T		A
gain/amplification	L			
normalization	L		–	
tremolo	L			LFO
violoning (attack smoothing)	L	T		A
Effects mainly on duration (D)				
time inversion	D	P,L,T	–	
time-scaling	D			
time-scaling with formant preservation	D		–	
time-scaling with vibrato preservation	D		–	
time-scaling with attack preservation	D		–	A
rhythm/swing change	D	T	–	A
Effects mainly on pitch (P)				
pitch-shifting without formant preservation	P	T		
pitch-shifting with formant preservation	P			
pitch change	P			A
pitch discretization (auto-tune)	P	T		A
harmonizer/smart harmony	P			A
(in-)harmonizer	P			A
Effects mainly on spatial aspects (S)				
distance change	S	L,T		
directivity	S	P,T		
Doppler effect	S	L,P		
echo	S	L		
granular delay	S	L,D,P,T		A
reverberation	S	L,D,T		
panning (2D, 3D)	S			
spectral panning	S	L,T		
rotary/Leslie	S	P,T		LFO

Table 1.2 Digital audio effects that mainly modify timbre only.

Effect name	Perceptual Attributes		RT	Control
	Main	Other		
Effects mainly on timbre (T)				
Effects on spectral envelope:				
filter	T	L		
arbitrary resolution filter	T	L		
comb filter	T	L,P		
resonant filter	T	L,P		
equalizer	T	L		
wah-wah	T	L,P		
auto-wah (sensitive wah)	T	L,D,P		LFO
envelope shifting	T	L		
envelope scaling	T	L		
envelope warping	T	L		
spectral centroid change	T	L		
Effects on phase:				
chorus	T			random
flanger	T	P		LFO
phaser	T	P		LFO
Effects on spectral structure:				
spectrum shifting	T	P		
adaptive ring modulation	T	P		A
texture change	T			
Effects on spectrum & envelope:				
distortion	T	L,P		
fuzz	T	L,P		
overdrive	T	L,P		
spectral (in-)harmonizer	T			
mutation	T	L,P		cross-A
spectral interpolation:	T	L,P		cross-A
vocoding	T	L,P		cross-A
cross-synthesis	T	L,P		cross-A
voice morphing	T	L,P		cross-A
timbral metamorphosis	T	L,P		cross-A
timbral morphing	T	L,P		cross-A
whispering/hoarseness	T	L	–	
de-esser	T	L		A
declicking	T	L	–	
denoising	T	L		
exciter	T	L		
enhancer	T	L		

Table 1.3 Digital audio effects that modify several perceptual attributes (on purpose).

Effect name	Perceptual Attributes		RT	Control
	Main	Other		
Effects modifying several perceptual attributes				
spectral compressor	L,T			
gender change	P,T	L		A
intonation change	L,P			A
martianisation	P,T	L		A
prosody change	L,D,P			A
resampling	D,T	L,P	–	
ring modulation	P,T			
robotization	P,T	L		
spectral tremolo	L,T	D		LFO
spectral warping	T,P	L		
time shuffling	L,D,P,T		–	
vibrato	L,P	T,D		LFO

By making use of heuristic maps [BB96] we can represent the various links between an effect and perceptual attributes, as depicted in Figure 1.4, where audio effects are linked in the center to the main perceptual attribute modified. Some sub-attributes (not necessarily perceptual) are introduced. For the sake of simplicity, audio effects are attached to the center only for the main modified perceptual attributes. When other attributes are slightly modified, they are indicated on the opposite side, i.e., at the figure bounds. When other perceptual attributes are slightly modified by an audio effect, those links are not connected to the center, in order to avoid overloading the heuristic map, but rather to the outer direction. A perceptual classification has the advantage of presenting audio effects according to the way they are perceived, taking into account the audible artifacts of the implementation techniques. The diagram in Figure 1.4, however, only represents each audio effect in its expected use (e.g., a compressor set to compress the dynamic range, which in turn slightly modifies the attacks and possibly timbre; it does not indicate all the possible settings, such as the attack smoothing and resulting timbral change when the attack time is set to 2s for instance). Of course, none of the presented classifications is perfect, and the adequacy of each depends on the goal we have in mind when using it. However, for sharing and spreading knowledge about audio effects between DSP programmers, musicians and listeners, this classification offers a vocabulary dealing with our auditory perception of the sound produced by the audio effect, that we all share since we all are listeners in the communication chain.

1.2.3 Interdisciplinary classification

Before introducing an interdisciplinary classification of audio effects that links the different layers of domain-specific descriptors, we recall sound effect classifications, as they provide clues for such interdisciplinary classifications. Sound effects have been thoroughly investigated in electroacoustic music. For instance, Schaeffer [Sch66] classified sounds according to: (i) *matter*, which is constituted of mass (noisiness; related to spectral density), harmonic timbre (harmonicity) and grain (the micro-structure of sound); (ii) *form*, which is constituted of dynamic (intensity evolution), and allure (e.g., frequency and amplitude modulation); (iii) *variation*, which is constituted of melodic profile (e.g., pitch variations) and mass profile (e.g., mass variations). In the context

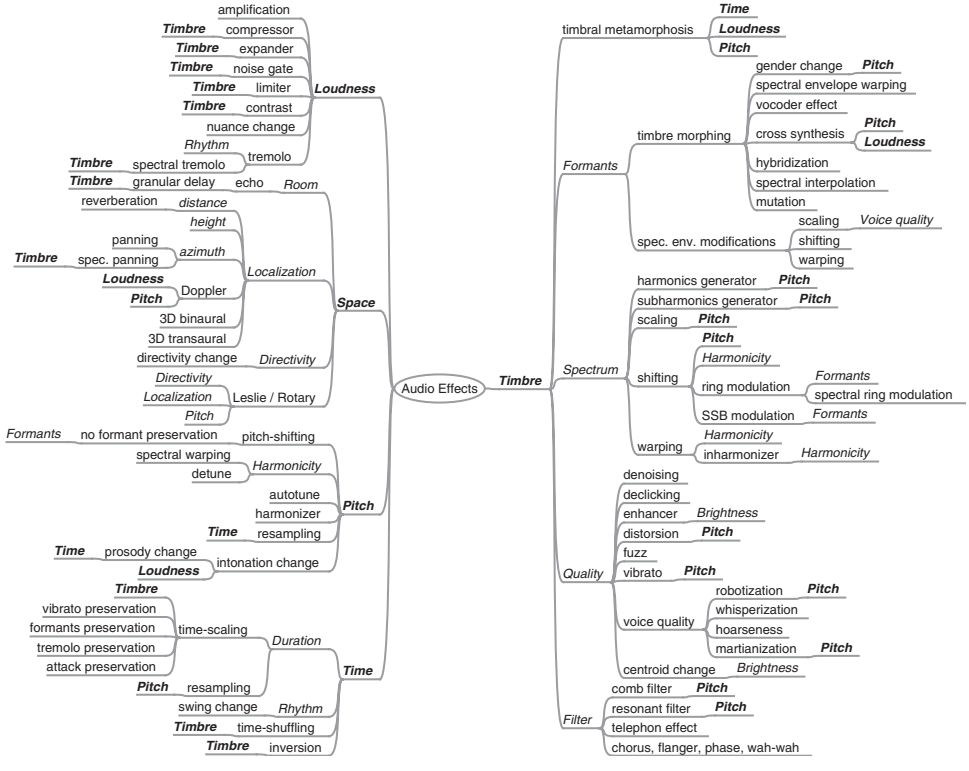

Figure 1.4 Perceptual classification of various audio effects. Bold-italic words are perceptual attributes (pitch, loudness, etc.). Italic words are perceptual sub-attributes (formants, harmonicity, etc.). Other words refer to the corresponding audio effects.

of ecological acoustics, Schafer [Sch77] introduced the idea that soundscapes reflect human activities. He proposed four main categories of environmental sounds: mechanical sounds (traffic and machines), human sounds (voices, footsteps), collective sounds (resulting from social activities) and sounds conveying information about the environment (warning signals or spatial effects). He considers four aspects of sounds: (i) emotional and affective qualities (aesthetics), (ii) function and meaning (semiotics and semantics), (iii) psychoacoustics (perception), (iv) acoustics (physical characteristics). That in turn can be used to develop classification categories [CKC+04]. Gaver [Gav93] also introduced the distinction between musical listening and everyday listening. Musical listening focuses on perceptual attributes of the sound itself (e.g., pitch, loudness), whereas everyday listening focuses on events to gather relevant information about our environment (e.g., car approaching), that is, not about the sound itself but rather about sound sources and actions producing sound. Recent research on soundscape perception validated this view by showing that people organize familiar sounds on the basis of source identification. But there is also evidence that the same sound can give rise to different cognitive representations which integrate semantic features (e.g., meaning attributed to the sound) into physical characteristics of the acoustic signal [GKP+05]. Therefore, semantic features must be taken into consideration when classifying sounds, but they cannot be matched with physical characteristics in a one-to-one relationship.

Similarly to sound effects, audio effects give rise to different semantic interpretations depending on how they are implemented or controlled. Semantic descriptors were investigated in the context of distortion [MM01] and different standpoints on reverberation were summarized in [Ble01]. An

interdisciplinary classification links the various layers of discipline-specific classifications ranging from low-level to high-level features as follows:

- Digital implementation technique
- Processing domain
- Applied processing
- Control type
- Perceptual attributes
- Semantic descriptors.

It is an attempt to bridge the gaps between discipline-specific classifications by extending previous research on isolated audio effects.

Chorus Revisited. The first example in Figure 1.5 concerns the chorus effect. As previously said, a chorus effect appears when several performers play together the same piece of music (same in melody, rhythm, dynamics) with the same kind of instrument. Slight pitch, dynamic, rhythm and timbre differences arise because the instruments are not physically identical, nor are perfectly tuned and synchronized. This effect provides some warmth to a sound, and can be considered as an effect on timbre: even though it performs slight modifications of pitch and time unfolding, the resulting effect is mainly on timbre. While its usual implementation involves one or many delay lines, with modulated length and controlled by a white noise, an alternative and more realistic sounding implementation consists in using several slightly pitch-shifted and time-scaled versions of the same sound with refined models (SOLA, phase vocoder, spectral models) and mixing them together. In this case, the resulting audio effect sounds more like a chorus of people or instruments playing the same harmonic and rhythmic patterns together. Therefore, this effect's control is a random generator (white noise), that controls a processing either in the time domain (using SOLA

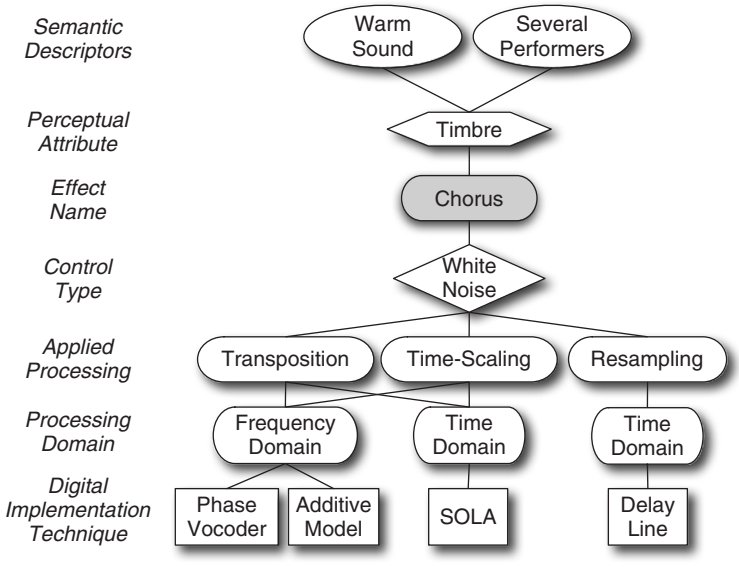

Figure 1.5 Transverse diagram for the chorus effect.

or a delay line), in the time-frequency domain (using the phase vocoder) or in the frequency domain (using spectral models).

Wah-Wah Revisited. The wah-wah is an effect that simulates vowel coarticulation. It can be implemented in the time domain using either a resonant filter or a series of resonant filters to simulate several formants of each vowels. In any case, these filters can be implemented in the time domain as well as in the time-frequency domain (phase vocoder) and in the frequency domain (with spectral models). From the usual wah-wah effect, variations can be derived by modifying its control. Figure 1.6 illustrates various control types for the wah-wah effect. With an LFO, the control is periodic and the wah-wah is called an "auto-wah." With gestural control, such as a foot pedal, it becomes the usual effect rock guitarists use since Jimmy Hendrix gave popularity to it. With an adaptive control based on the attack of each note, it becomes a "sensitive wah" that moves from "a" at the attack to "u" during the release. We now can better see the importance of specifying the control type as part of the effect definition.

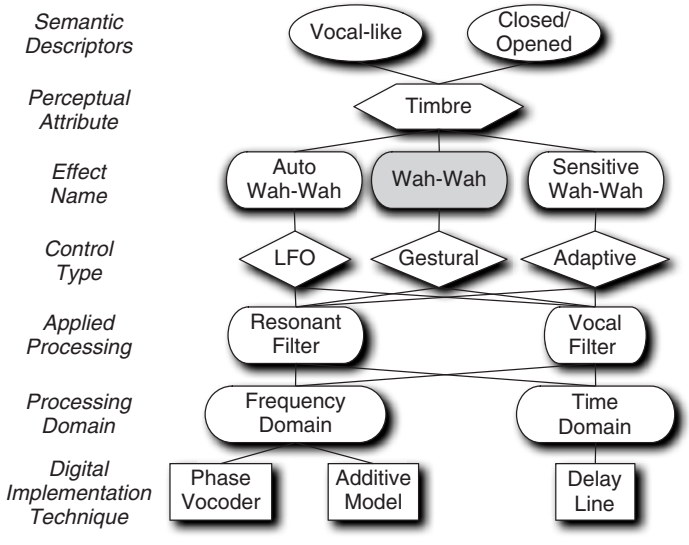

Figure 1.6 Transverse diagram for the wah-wah effect: the control type defines the effect's name, i.e., wah-wah, automatic wah-wah (with LFO) or sensitive wah-wah (adaptive control).

Comb Filter Revisited. Figure 1.7 depicts the interdisciplinary classification for the comb filter. This effect corresponds to filtering a signal using a comb-shaped frequency response. When the signal is rich and contains either a lot of partials, or a certain amount of noise, its filtering gives rise to a timbral pitch that can easily be heard. The sound is then similar to a sound heard through the resonances of a tube, or even vocal formants when the tube length is properly adjusted. As any filter, the effect can be implemented in both the time domain (using delay lines), the time-frequency domain (phase vocoder) and the frequency domain (spectral models). When controlled with a LFO, the comb filter changes its name to "phasing," which sounds similar to a plane landing, and has been used in songs during the late 1960s to simulate the effects of drugs onto perception.

Cross-synthesis revisited. The transverse diagram for cross-synthesis shown in Figure 1.8 consists in applying the time-varying spectral envelope of one sound onto the source of a second sound, after having separated their two source and filter components. Since this effect takes the whole spectral envelope of one sound, it also conveys some amplitude and time information, resulting in modifications of timbre, but also loudness, and time and rhythm. It may provide the

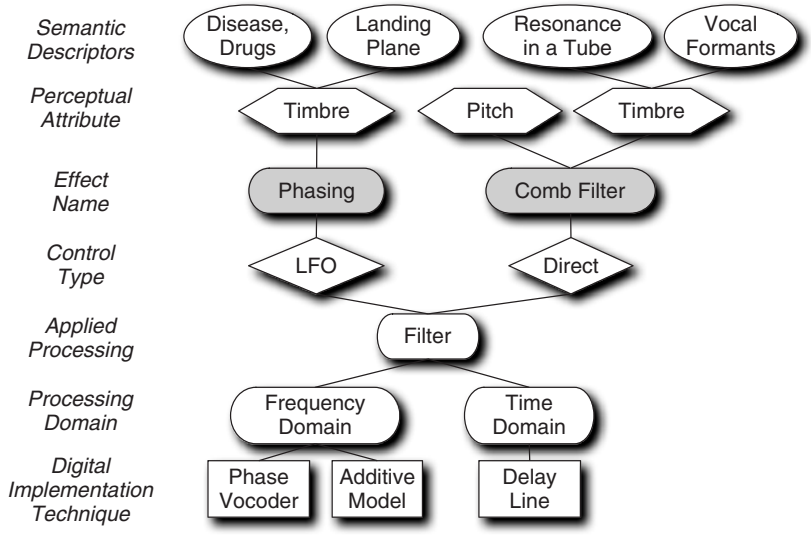

Figure 1.7 Transverse diagram for the comb-filter effect: a modification of the control by adding a LFO results in another effect called "phasing."

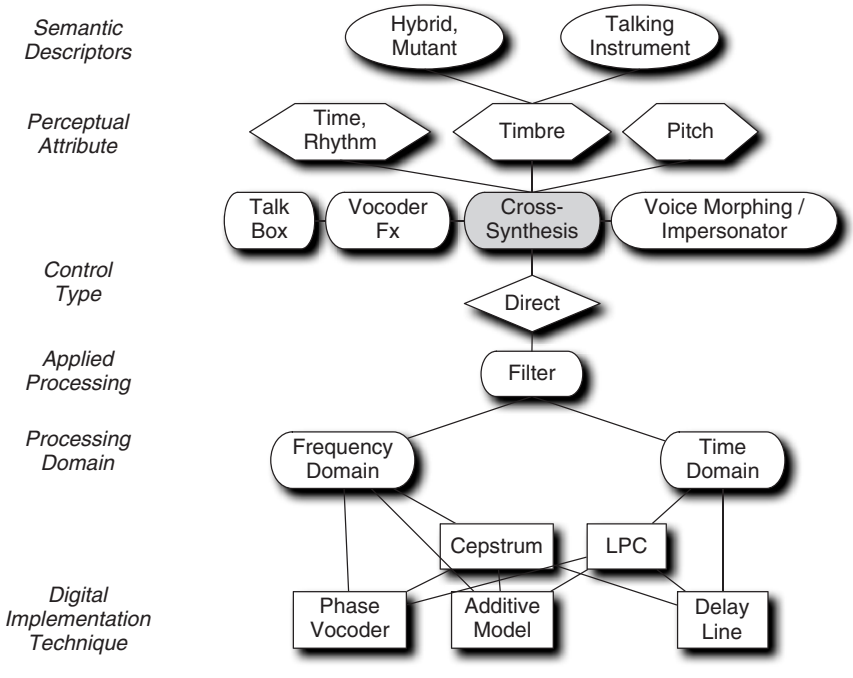

Figure 1.8 Transverse diagram for the cross-synthesis effect.

illusion of a talking instrument when the resonances of a human voice are applied onto the source of a musical instrument. It then provides a hybrid, mutant voice. After the source-filter separation, the filtering of the source of sound A with the filter from sound B can be applied in the time domain as well as in the frequency and the time-frequency domains. Other perceptually similar effects on voice are called voice morphing (as the processing used to produce the castrato's voice in the movie Farinelli's soundtrack), "voice impersonator" (the timbre of a voice from the database is mapped to your singing voice in real time), the "vocoder effect" (based on the classical vocoder), or the "talk box" (where the filter of a voice is applied to a guitar sound without removing its original resonances, then adding the voice's resonances to the guitar's resonances; as in Peter Frampton's famous "Do you feel like I do").

Distortion revisited. A fifth example is the distortion effect depicted in Figure 1.9. Distortion is produced from a soft or hard clipping of the signal, and results in a harmonic enrichment of a sound. It is widely used in popular music, especially through electric guitar that conveyed it from the beginning, due to amplification. Distortions can be implemented using amplitude warping (e.g., with Chebyshev polynomials or wave shaping), or with physical modeling of valve amplifiers. Depending on its settings, it may provide a warm sound, an aggressive sound, a bad quality sound, a metallic sound, and so on.

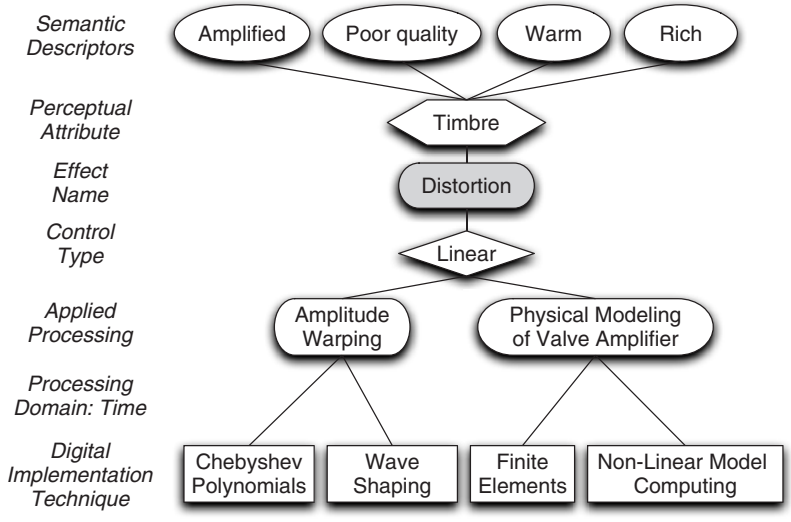

Figure 1.9 Transverse diagram for the distortion effect.

Equalizer revisited. A last example is the equalizer depicted in Figure 1.10. Its design consists of a series of shelving and peak filters that can be implemented in the time domain (filters), in the time-frequency domain (phase vocoder) or in the frequency domain (with spectral models). The user directly controls the gain, bandwidth and center frequency in order to apply modifications of the energy in each frequency band, in order to better suit aesthetic needs and also correct losses in the transducer chain.

We illustrated and summarized various classifications of audio effects elaborated in different disciplinary fields. An interdisciplinary classification links the different layers of domain-specific features and aims to facilitate knowledge exchange between the fields of musical acoustics, signal processing, psychoacoustics and cognition. Besides addressing the classification of audio effects, we further explained the relationships between structural and control parameters of signal processing algorithms and the perceptual attributes modified by audio effects. A generalization of this

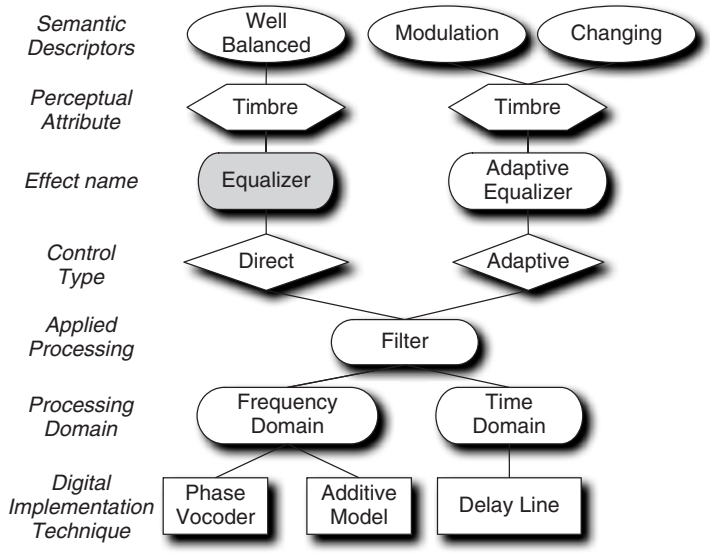

Figure 1.10 Transverse diagram for the equalizer effect.

classification to all audio effects would have a strong impact on pedagogy, knowledge sharing across disciplinary fields and musical practice. For example, DSP engineers conceive better tools when they know how it can be used in a musical context. Furthermore, linking perceptual features to signal processing techniques enables the development of more intuitive user interfaces providing control over high-level perceptual and cognitive attributes rather than low-level signal parameters.

1.3 Fundamentals of digital signal processing

The fundamentals of digital signal processing consist of the description of *digital signals* – in the context of this book we use digital audio signals – as a sequence of numbers with appropriate number representation and the description of *digital systems*, which are described by software algorithms to calculate an output sequence of numbers from an input sequence of numbers. The visual representation of digital systems is achieved by functional block diagram representations or signal flow graphs. We will focus on some simple basics as an introduction to the notation and refer the reader to the literature for an introduction to digital signal processing [ME93, Orf96, Zöl05, MSY98, Mit01].

1.3.1 Digital signals

The digital signal representation of an analog audio signal as a sequence of numbers is achieved by an analog-to-digital converter (ADC). The ADC performs *sampling* of the amplitudes of the analog signal $x(t)$ on an equidistant grid along the horizontal time axis and *quantization* of the amplitudes to fixed samples represented by numbers $x(n)$ along the vertical amplitude axis (see Fig. 1.11). The samples are shown as vertical lines with dots on the top. The analog signal $x(t)$ denotes the signal amplitude over continuous time t in micro seconds. Following the ADC, the digital (discrete time and quantized amplitude) signal is a sequence (stream) of samples $x(n)$ represented by numbers over the discrete time index n. The time distance between two consecutive samples is termed *sampling interval* T (sampling period) and the reciprocal is the *sampling frequency*

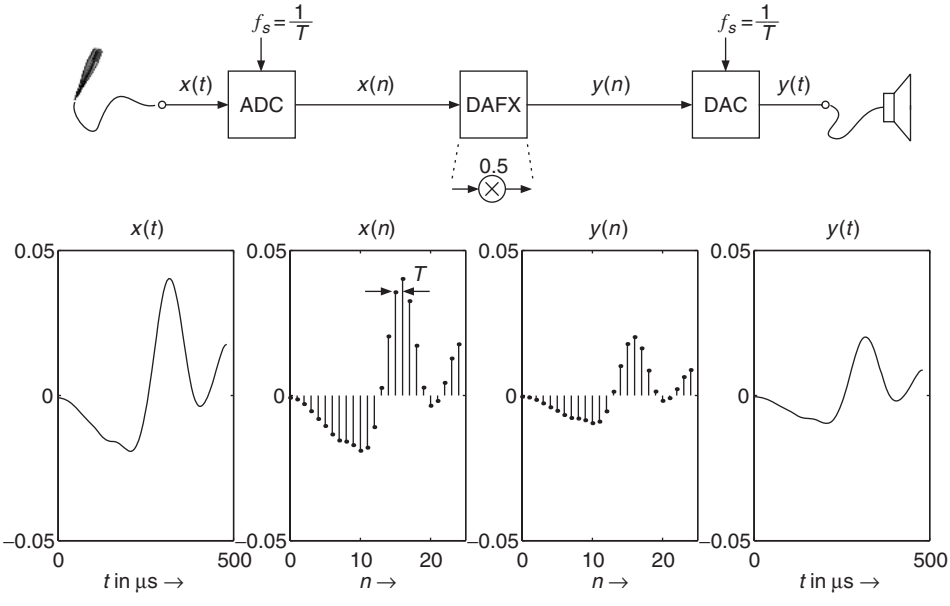

Figure 1.11 Sampling and quantizing by ADC, digital audio effects and reconstruction by DAC.

$f_S = 1/T$ (sampling rate). The sampling frequency reflects the number of samples per second in Hertz (Hz). According to the sampling theorem, it has to be chosen as twice the highest frequency $f_{\max}$ (signal bandwidth) contained in the analog signal, namely $f_S > 2 \cdot f_{\max}$. If we are forced to use a fixed sampling frequency f_S, we have to make sure that our input signal to be sampled has a bandwidth according to $f_{\max} = f_S/2$. If not, we have to reject higher frequencies by filtering with a lowpass filter which only passes all frequencies up to $f_{\max}$. The digital signal is then passed to a DAFX box (digital system), which in this example performs a simple multiplication of each sample by 0.5 to deliver the output signal $y(n) = 0.5 \cdot x(n)$. This signal $y(n)$ is then forwarded to a digital-to-analog converter DAC, which reconstructs the analog signal $y(t)$. The output signal $y(t)$ has half the amplitude of the input signal $x(t)$.

Figure 1.12 shows some digital signals to demonstrate different graphical representations (see M-file 1.1). The upper part shows 8000 samples, the middle part the first 1000 samples and the lower part shows the first 100 samples out of a digital audio signal. Only if the number of samples inside a figure is sufficiently low, will the line with dot graphical representation be used for a digital signal.

M-file 1.1 (figure1_03.m)

```
% Author: U. Zölzer
[x,FS,NBITS]=wavread('ton2.wav');

figure(1)
subplot(3,1,1);
plot(0:7999,x(1:8000));ylabel('x(n)');
subplot(3,1,2);
plot(0:999,x(1:1000));ylabel('x(n)');
subplot(3,1,3);
stem(0:99,x(1:100),'.');ylabel('x(n)');
xlabel('n \rightarrow');
```

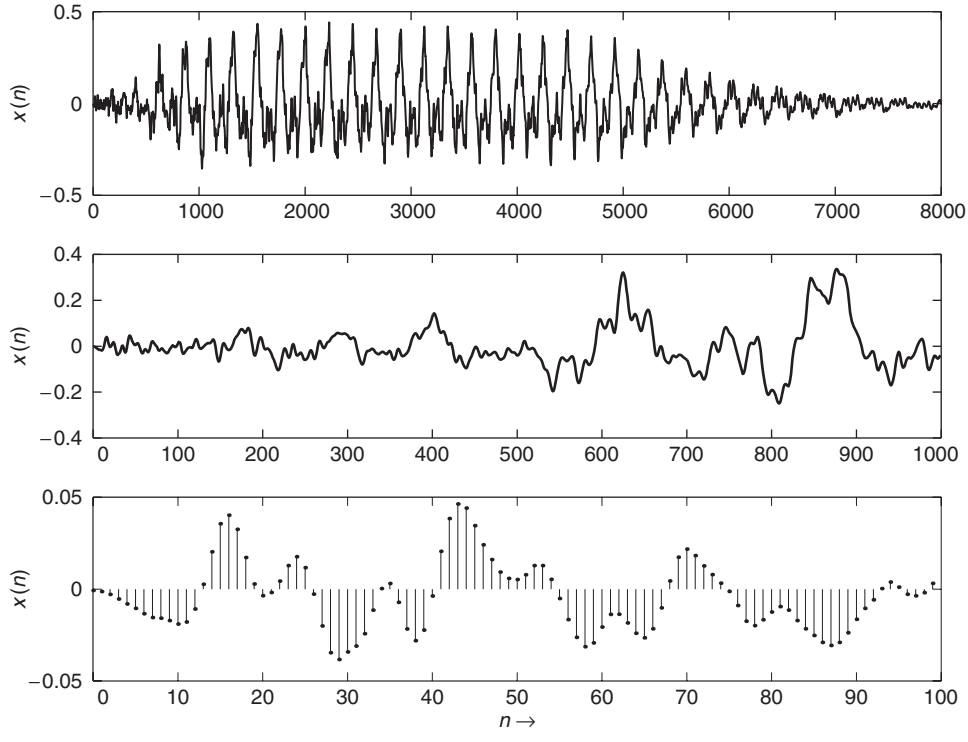

Figure 1.12 Different time representations for digital audio signals.

Two different vertical scale formats for digital audio signals are shown in Figure 1.13. The quantization of the amplitudes to fixed numbers in the range $-32768 \ldots 32767$ is based on a 16-bit representation of the sample amplitudes which allows 2^{16} quantized values in the range $-2^{15} \ldots 2^{15} - 1$. For a general w-bit representation the number range is $-2^{w-1} \ldots 2^{w-1} - 1$. This representation is called the integer number representation. If we divide all integer numbers by the maximum absolute value, for example 32768, we come to the normalized vertical scale in Figure 1.13, which is in the range $-1 \ldots 1 - Q$, where Q is the quantization step size. It and can be calculated by $Q = 2^{-(w-1)}$, which leads to $Q = 3.0518 \times 10^{-5}$ for $w = 16$. Figure 1.13 also displays the horizontal scale formats, namely the continuous-time axis, the discrete-time axis and the normalized discrete-time axis, which will be used normally. After this narrow description we can define a digital signal as a discrete-time and discrete-amplitude signal, which is formed by sampling an analog signal and by quantization of the amplitude onto a fixed number of amplitude values. The digital signal is represented by a sequence of numbers $x(n)$. Reconstruction of analog signals can be performed by DACs. Further details of ADCs and DACs and the related theory can be found in the literature. For our discussion of digital audio effects this short introduction to digital signals is sufficient.

Signal processing algorithms usually process signals by either *block processing* or *sample-by-sample processing*. Examples for digital audio effects are presented in [Arf98]. For block processing, samples are first collected in a memory buffer and then processed each time the buffer is completely filled with new data. Examples of such algorithms are *fast Fourier transforms (FFTs)* for spectra computations and *fast convolution*. In sample processing algorithms, each input sample is processed on a sample-by-sample basis.

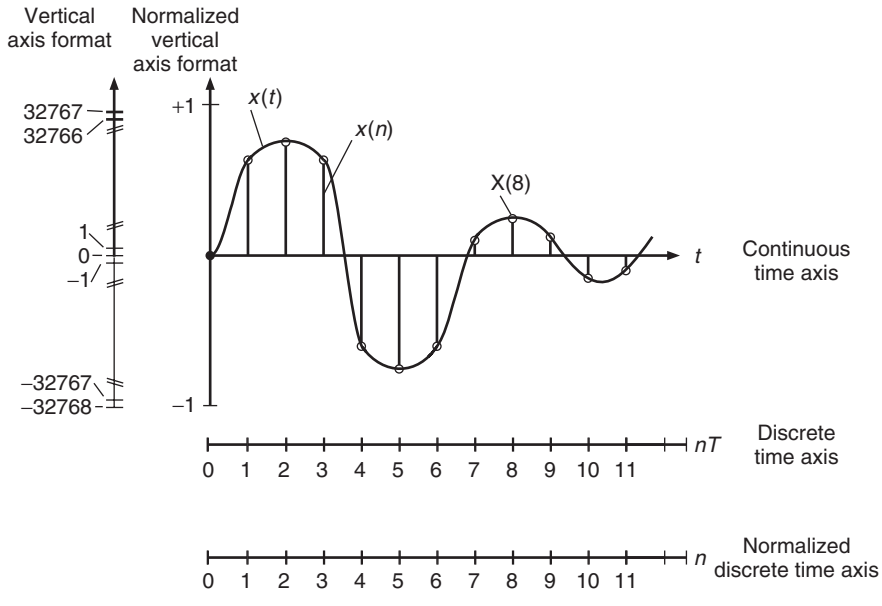

Figure 1.13 Vertical and horizontal scale formats for digital audio signals.

A basic algorithm for weighting of a sound $x(n)$ (see Figure 1.11) by a constant factor a demonstrates a sample-by-sample processing (see M-file 1.2). The input signal is represented by a vector of numbers `[x(1), x(2),..., x(length(x))]`.

M-file 1.2 (`sbs_alg.m`)

```
% Author: U. Zölzer
% Read input sound file into vector x(n) and sampling frequency FS
[x,FS]=wavread('input filename');
% Sample-by sample algorithm y(n)=a*x(n)
for n=1:length(x),
     y(n)=a * x(n);
end;
% Write y(n) into output sound file with number of
% bits Nbits and sampling frequency FS
wavwrite(y,FS,Nbits,'output filename');
```

1.3.2 Spectrum analysis of digital signals

The spectrum of a signal shows the distribution of energy over the frequency range. The upper part of Figure 1.14 shows the spectrum of a short time slot of an analog audio signal. The frequencies range up to 20 kHz. The sampling and quantization of the analog signal with sampling frequency of $f_S = 40$ kHz leads to a corresponding digital signal. The spectrum of the digital signal of the same time slot is shown in the lower part of Figure 1.14. The sampling operation leads to a replication of the baseband spectrum of the analog signal [Orf96]. The frequency contents from 0 Hz up to 20 kHz of the analog signal now also appear from 40 kHz up to 60 kHz and the folded version of it from 40 kHz down to 20 kHz. The replication of this first image of the baseband spectrum at 40 kHz will now also appear at integer multiples of the sampling frequency of $f_S = 40$ kHz. But

notice that the spectrum of the digital signal from 0 up to 20 kHz shows exactly the same shape as the spectrum of the analog signal. The reconstruction of the analog signal out of the digital signal is achieved by simply lowpass filtering the digital signal, rejecting frequencies higher than $f_S/2 = 20$ kHz. If we consider the spectrum of the digital signal in the lower part of Fig. 1.14 and reject all frequencies higher than 20 kHz, we come back to the spectrum of the analog signal in the upper part of the figure.

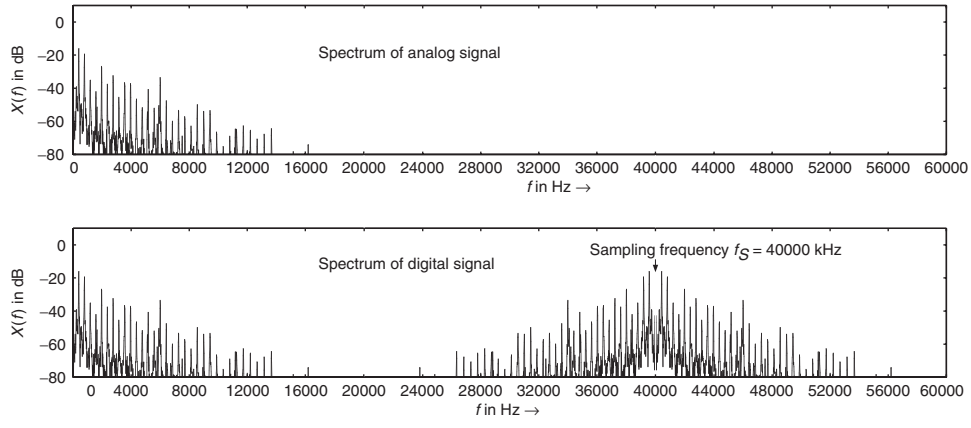

Figure 1.14 Spectra of analog and digital signals.

Discrete Fourier transform

The spectrum of a digital signal can be computed by the discrete Fourier transform (DFT) which is given by

$$X(k) = \mathrm{DFT}[x(n)] = \sum_{n=0}^{N-1} x(n)\mathrm{e}^{-j2\pi nk/N} \quad k = 0, 1, \ldots, N-1. \tag{1.1}$$

The fast version of the above formula is called the fast Fourier transform (FFT). The FFT takes N consecutive samples out of the signal $x(n)$ and performs a mathematical operation to yield N samples $X(k)$ of the spectrum of the signal. Figure 1.15 demonstrates the results of a 16-point FFT applied to 16 samples of a cosine signal. The result is normalized by N according to `X=abs(fft(x,N))/N;`.

The N samples $X(k) = X_R(k) + jX_I(k)$ are complex-valued with a real part $X_R(k)$ and an imaginary part $X_I(k)$ from which one can compute the absolute value

$$|X(k)| = \sqrt{X_R^2(k) + X_I^2(k)} \quad k = 0, 1, \ldots, N-1 \tag{1.2}$$

which is the magnitude spectrum, and the phase

$$\varphi(k) = \arctan \frac{X_I(k)}{X_R(k)} \quad k = 0, 1, \ldots, N-1 \tag{1.3}$$

which is the phase spectrum. Figure 1.15 also shows that the FFT algorithm leads to N equidistant frequency points which give N samples of the spectrum of the signal starting from 0 Hz in steps of $\frac{f_S}{N}$ up to $\frac{N-1}{N} f_S$. These frequency points are given by $k\frac{f_S}{N}$, where k is running from $0, 1, 2, \ldots, N-1$. The magnitude spectrum $|X(f)|$ is often plotted over a logarithmic amplitude scale

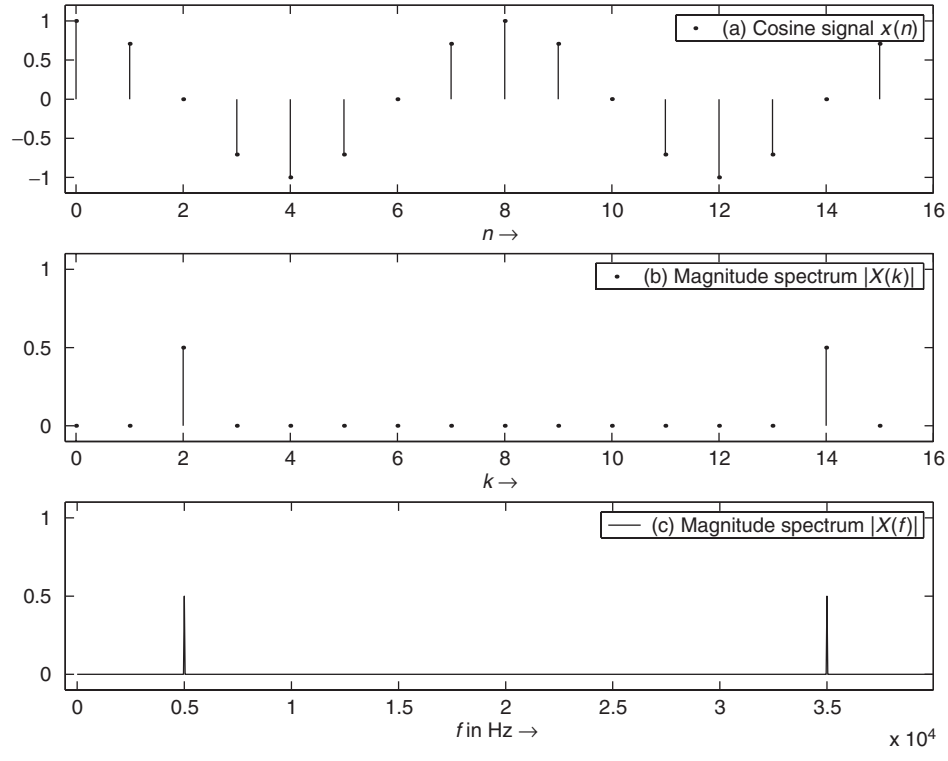

Figure 1.15 Spectrum analysis with FFT algorithm: (a) digital cosine with $N = 16$ samples, (b) magnitude spectrum $|X(k)|$ with $N = 16$ frequency samples and (c) magnitude spectrum $|X(f)|$ from 0 Hz up to the sampling frequency $f_S = 40\,000$ Hz.

according to $20 \log_{10}\left(\frac{X(f)}{0.5}\right)$ which gives 0 dB for a sinusoid of maximum amplitude ± 1. This normalization is equivalent to $20 \log_{10}\left(\frac{X(k)}{N/2}\right)$. Figure 1.16 shows this representation of the example from Fig. 1.15. Images of the baseband spectrum occur at the sampling frequency f_S and multiples of f_S. Therefore we see the original frequency at 5 kHz and in the first image spectrum the folded frequency $f_S - f_{\text{cosine}} = 40\,000$ Hz $- 5000$ Hz $= 35\,000$ Hz.

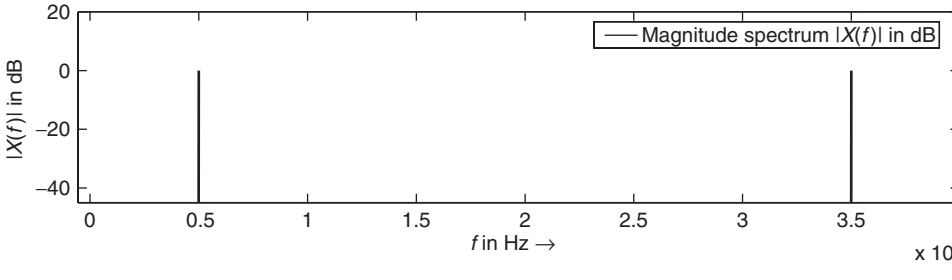

Figure 1.16 Magnitude spectrum $|X(f)|$ in dB from 0 Hz up to the sampling frequency $f_S = 40000$ Hz.

Inverse discrete Fourier transform (IDFT)

Whilst the DFT is used as the transform from the discrete-time domain to the discrete-frequency domain for spectrum analysis, the inverse discrete Fourier transform (IDFT) allows the transform from the discrete-frequency domain to the discrete-time domain. The IDFT algorithm is given by

$$x(n) = \text{IDFT}[X(k)] = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\mathrm{e}^{j2\pi nk/N} \quad n = 0, 1, \ldots, N-1. \tag{1.4}$$

The fast version of the IDFT is called the inverse Fast Fourier transform (IFFT). Taking N complex-valued numbers with the property $X(k) = X^*(N-k)$ in the frequency domain and then performing the IFFT gives N discrete-time samples $x(n)$, which are real-valued.

Frequency resolution: zero-padding and window functions

To increase the frequency resolution for spectrum analysis we simply take more samples for the FFT algorithm. Typical numbers for the FFT resolution are $N = 256$, 512, 1024, 2048, 4096 and 8192. If we are only interested in computing the spectrum of 64 samples and would like to increase the frequency resolution from $f_S/64$ to $f_S/1024$, we have to extend the sequence of 64 audio samples by adding zero samples up to the length 1024 and then perform an 1024-point FFT. This technique is called zero-padding and is illustrated in Figure 1.17 and by M-file 1.3. The upper left

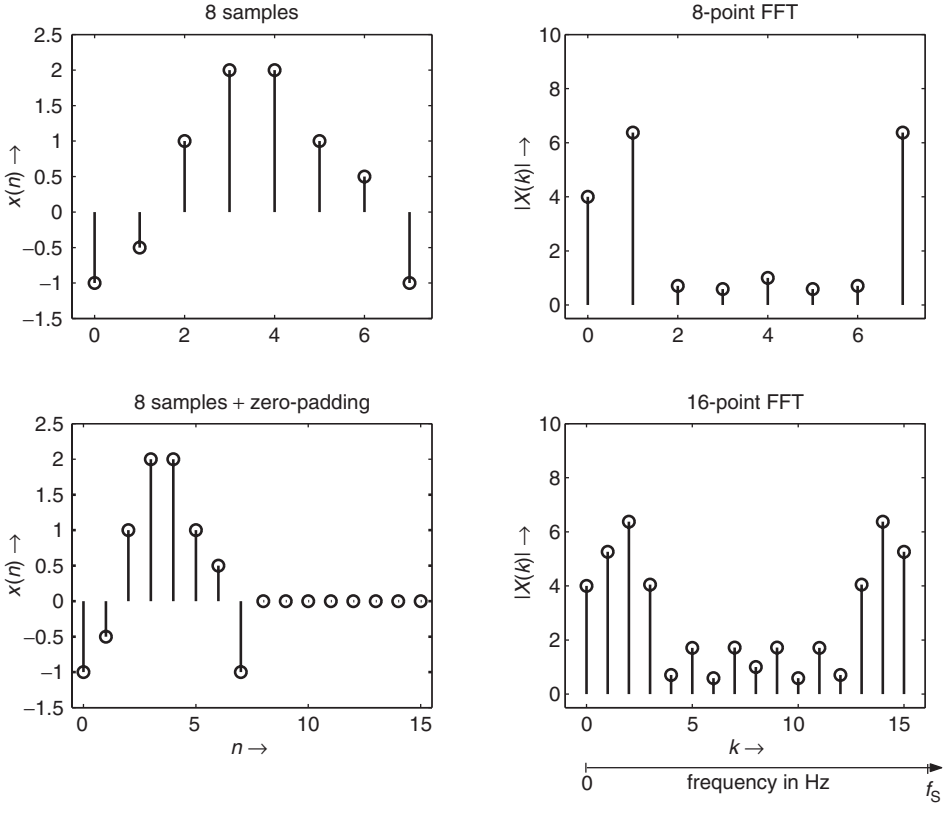

Figure 1.17 Zero-padding to increase frequency resolution.

part shows the original sequence of eight samples and the upper right part shows the corresponding eight-point FFT result. The lower left part illustrates the adding of eight zero samples to the original eight-sample sequence up to the length $N = 16$. The lower right part illustrates the magnitude spectrum $|X(k)|$ resulting from the 16-point FFT of the zero-padded sequence of length $N = 16$. Notice the increase in frequency resolution between the eight-point and 16-point FFT. Between each frequency bin of the upper spectrum a new frequency bin in the lower spectrum is calculated. Bins $k = 0, 2, 4, 6, 8, 10, 12, 14$ of the 16-point FFT correspond to bins $k = 0, 1, 2, 3, 4, 5, 6, 7$ of the eight-point FFT. These N frequency bins cover the frequency range from 0 Hz up to $\frac{N-1}{N} f_S$.

M-file 1.3 (`figure1_17.m`)

```
%Author: U. Zölzer
x1=[-1 -0.5 1 2 2 1 0.5 -1];
x2(16)=0;
x2(1:8)=x1;

subplot(221);
stem(0:1:7,x1);axis([-0.5 7.5 -1.5 2.5]);
ylabel('x(n) \rightarrow');title('8 samples');
subplot(222);
stem(0:1:7,abs(fft(x1)));axis([-0.5 7.5 -0.5 10]);
ylabel('|X(k)| \rightarrow');title('8-point FFT');

subplot(223);
stem(0:1:15,x2);axis([-0.5 15.5 -1.5 2.5]);
xlabel('n \rightarrow');ylabel('x(n) \rightarrow');
title('8 samples+zero-padding');

subplot(224);
stem(0:1:15,abs(fft(x2)));axis([-1 16 -0.5 10]);
xlabel('k \rightarrow');ylabel('|X(k)| \rightarrow');
title('16-point FFT');
```

The leakage effect occurs due to cutting out N samples from the signal. This effect is shown in the upper part of Figure 1.18 and demonstrated by the corresponding M-file 1.4. The cosine spectrum is smeared around the frequency. We can reduce the leakage effect by selecting a window function like Blackman window and Hamming window

$$w_B(n) = 0.42 - 0.5\cos(2\pi n/N) + 0.08\cos(4\pi n/N), \tag{1.5}$$

$$w_H(n) = 0.54 - 0.46\cos(2\pi n/N) \tag{1.6}$$

$$n = 0, 1, \ldots N - 1.$$

and weighting the N audio samples by the window function. This weighting is performed according to $x_w = w(n) \cdot x(n) / \left(\sum_k w(k)\right)$ with $0 \le n \le N - 1$ and then an FFT of the weighted signal is performed. The cosine weighted by a window and the corresponding spectrum is shown in the middle part of Figure 1.18. The lower part of Figure 1.18 shows a segment of an audio signal weighted by the Blackman window and the corresponding spectrum via a FFT. Figure 1.19 shows further simple examples for the reduction of the leakage effect and can be generated by the M-file 1.5.

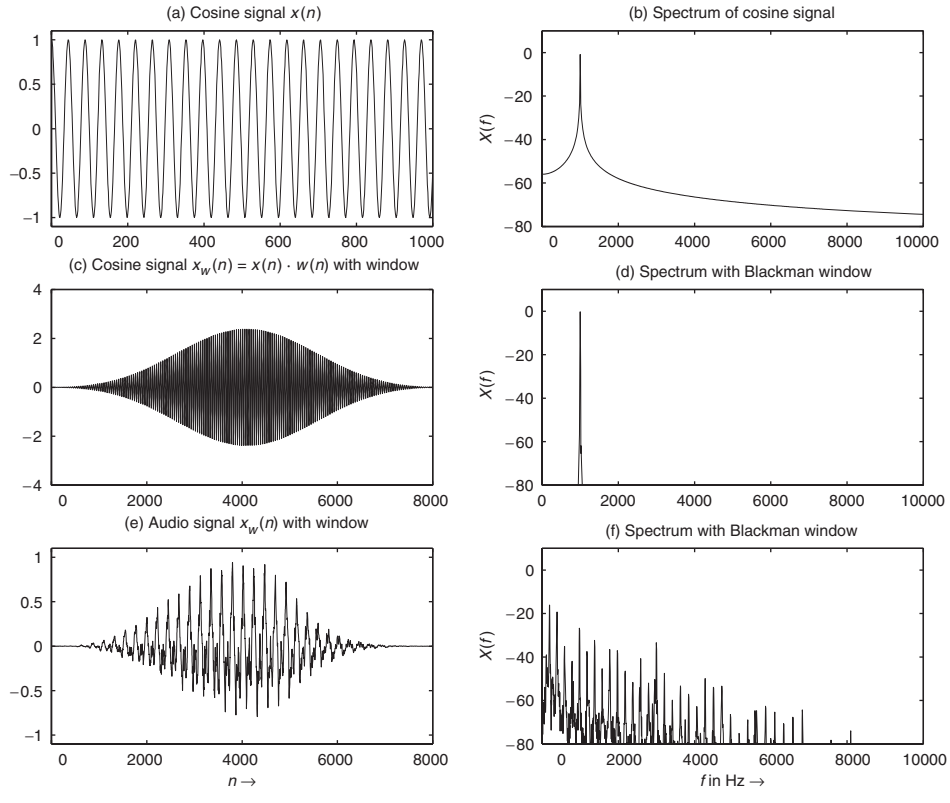

Figure 1.18 Spectrum analysis of digital signals: take N audio samples and perform an N point discrete Fourier transform to yield N samples of the spectrum of the signal starting from 0 Hz over $k\frac{f_S}{N}$ where k is running from $0, 1, 2, \ldots, N-1$. For (a)–(d), $x(n) = \cos(2 \cdot \pi \cdot \frac{1\ kHz}{44.1\ kHz} \cdot n)$.

M-file 1.4 (figure1_18.m)

```
%Author: U. Zölzer
x=cos(2*pi*1000*(0:1:N-1)/44100)';
figure(2)
W=blackman(N);
W=N*W/sum(W); % scaling of window
f=((0:N/2-1)/N)*FS;

xw=x.*W;
subplot(3,2,1);plot(0:N-1,x);
axis([0 1000 -1.1 1.1]);
title('a) Cosine signal x(n)')

subplot(3,2,3);plot(0:N-1,xw);axis([0 8000 -4 4]);
title('c) Cosine signal x_w(n)=x(n) \cdot w(n) with window')

X=20*log10(abs(fft(x,N))/(N/2));
subplot(3,2,2);plot(f,X(1:N/2));
axis([0 10000 -80 10]);
```

```
ylabel('X(f)');
title('b) Spectrum of cosine signal')

Xw=20*log10(abs(fft(xw,N))/(N/2));
subplot(3,2,4);plot(f,Xw(1:N/2));
axis([0 10000 -80 10]);
ylabel('X(f)');
title('d) Spectrum with Blackman window')

s=u1(1:N).*W;
subplot(3,2,5);plot(0:N-1,s);axis([0 8000 -1.1 1.1]);
xlabel('n \rightarrow');
title('e) Audio signal x_w(n) with window')

Sw=20*log10(abs(fft(s,N))/(N/2));
subplot(3,2,6);plot(f,Sw(1:N/2));
axis([0 10000 -80 10]);
ylabel('X(f)');
title('f) Spectrum with Blackman window')
xlabel('f in Hz \rightarrow');
```

M-file 1.5 (figure1_19.m)

```
%Author: U. Zölzer
x=[-1 -0.5 1 2 2 1 0.5 -1];
w=blackman(8);
w=w*8/sum(w);
x1=x.*w';
x2(16)=0;
x2(1:8)=x1;

subplot(421);
stem(0:1:7,x);axis([-0.5 7.5 -1.5 2.5]);
ylabel('x(n) \rightarrow');
title('a) 8 samples');
subplot(423);
stem(0:1:7,w);axis([-0.5 7.5 -1.5 3]);
ylabel('w(n) \rightarrow');
title('b) 8 samples Blackman window');

subplot(425);
stem(0:1:7,x1);axis([-0.5 7.5 -1.5 6]);
ylabel('x_w(n) \rightarrow');
title('c) x(n)\cdot w(n)');

subplot(427);
stem(0:1:15,x2);axis([-0.5 15.5 -1.5 6]);
xlabel('n \rightarrow');ylabel('x_w(n) \rightarrow');
title('d) x(n)\cdot w(n) + zero-padding');

subplot(222);
stem(0:1:7,abs(fft(x1)));axis([-0.5 7.5 -0.5 15]);
ylabel('|X(k)| \rightarrow');
title('8-point FFT of c)');
```

```
subplot(224);
stem(0:1:15,abs(fft(x2)));axis([-1 16 -0.5 15]);
xlabel('k \rightarrow');ylabel('|X(k)| \rightarrow');
title('16-point FFT of d)');
```

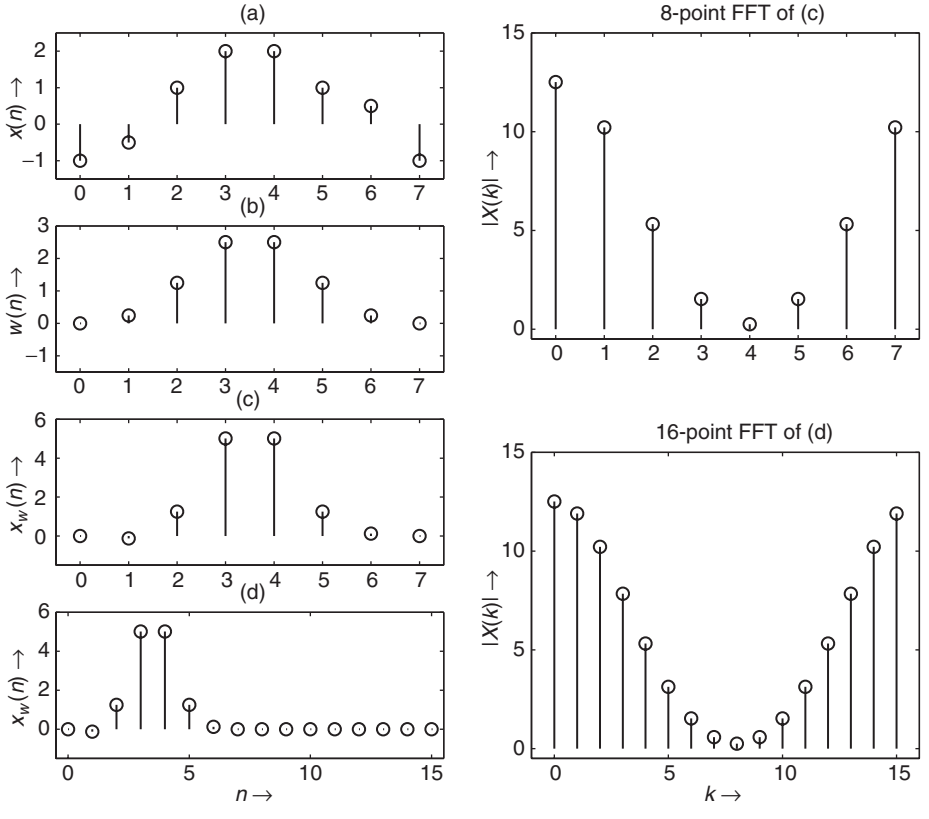

Figure 1.19 Reduction of the leakage effect by window functions: (a) the original signal, (b) the Blackman window function of length $N = 8$, (c) product $x(n) \cdot w_B(n)$ with $0 \leq n \leq N-1$, (d) zero-padding applied to $x(n) \cdot w_B(n)$ up to length $N = 16$ and the corresponding spectra are shown on the right side.

Spectrogram: time-frequency representation

A special time-frequency representation is the spectrogram which gives an estimate of the short-time, time-localized frequency content of the signal. To obtain the spectrogram the signal is split into segments of length N, which are multiplied by a window and an FFT is performed (see Figure 1.20). To increase the time-localization of the short-time spectra an overlap of the weighted segments can be used. A special visual representation of the short-time spectra is the spectrogram in Figure 1.21. Time increases linearly across the horizontal axis and frequency increases across the vertical axis. So each vertical line represents the absolute value $|X(f)|$ over frequency by a

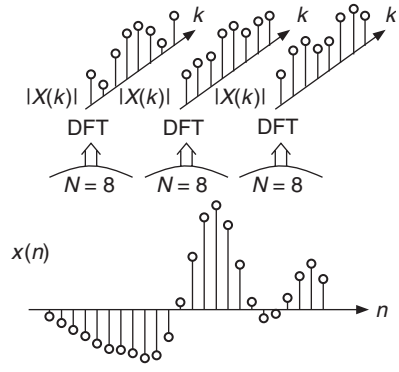

Figure 1.20 Short-time spectrum analysis by FFT.

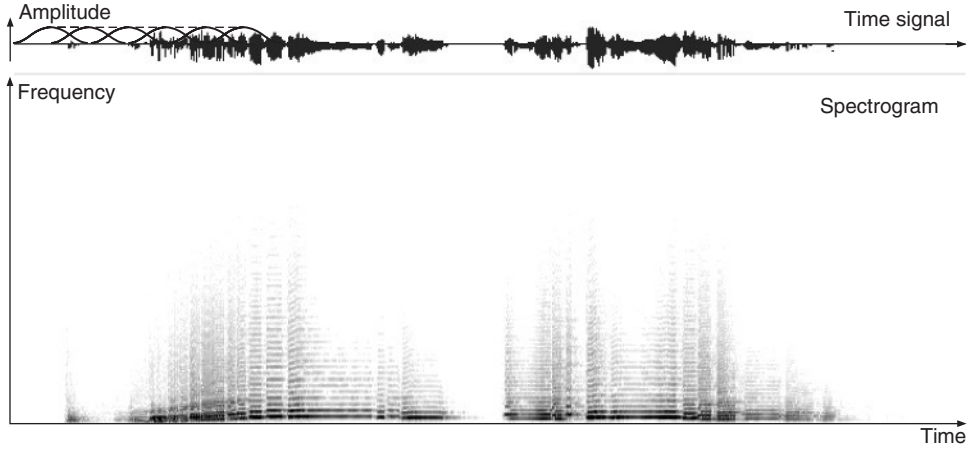

Figure 1.21 Spectrogram via FFT of weighted segments.

grey-scale value (see Figure 1.21). Only frequencies up to half the sampling frequency are shown. The calculation of the spectrogram from a signal can be performed by the **MATLAB** function `B = specgram(x,nFFT,Fs,window,nOverlap)`.

Another time-frequency representation of the short-time Fourier transforms of a signal $x(n)$ is the waterfall representation in Figure 1.22, which can be produced by M-file 1.6 that calls the waterfall computation algorithm given by M-file 1.7.

M-file 1.6 (`figure1_22.m`)

```
%Author: U. Zölzer
[signal,FS,NBITS]=wavread('ton2');
subplot(211);plot(signal);
subplot(212);
waterfspec(signal,256,256,512,FS,20,-100);
```

M-file 1.7 (waterfspec.m)

```
function yy=waterfspec(signal,start,steps,N,fS,clippingpoint,baseplane)
% Authors: J. Schattschneider, U. Zölzer
% waterfspec( signal, start, steps, N, fS, clippingpoint, baseplane)
%
% shows short-time spectra of signal, starting
% at k=start, with increments of STEP with N-point FFT
% dynamic range from -baseplane in dB up to 20*log(clippingpoint)
% in dB versus time axis
echo off;
if nargin<7, baseplane=-100; end
if nargin<6, clippingpoint=0; end
if nargin<5, fS=48000; end
if nargin<4, N=1024; end         % default FFT
if nargin<3, steps=round(length(signal)/25); end
if nargin<2, start=0; end

windoo=blackman(N);              % window - default
windoo=windoo*N/sum(windoo);     % scaling
% Calculation of number of spectra nos
  n=length(signal);
  rest=n-start-N;
  nos=round(rest/steps);
  if nos>rest/steps, nos=nos-1; end
% vectors for 3D representation
  x=linspace(0, fS/1000 ,N+1);
  z=x-x;
  cup=z+clippingpoint;
  cdown=z+baseplane;

  signal=signal+0.0000001;
% Computation of spectra and visual representation
  for i=1:1:nos,
    spek1=20.*log10(abs(fft(windoo.*signal(1+start+....
    ....i*steps:start+N+i*steps))./(N)/0.5);
    spek=[-200 ; spek1(1:N)];
    spek=(spek>cup').*cup'+(spek<=cup').*spek;
    spek=(spek<cdown').*cdown'+(spek>=cdown').*spek;
    spek(1)=baseplane-10;
    spek(N/2)=baseplane-10;
    y=x-x+(i-1);
    if i==1,
       p=plot3(x(1:N/2),y(1:N/2),spek(1:N/2),'k');
       set(p,'Linewidth',0.1);
    end
       pp=patch(x(1:N/2),y(1:N/2),spek(1:N/2),'w','Visible','on');
       set(pp,'Linewidth',0.1);
   end;
set(gca,'DrawMode','fast');
axis([-0.3 fS/2000+0.3 0 nos baseplane-10 0]);
set(gca,'Ydir','reverse');
view(12,40);
```

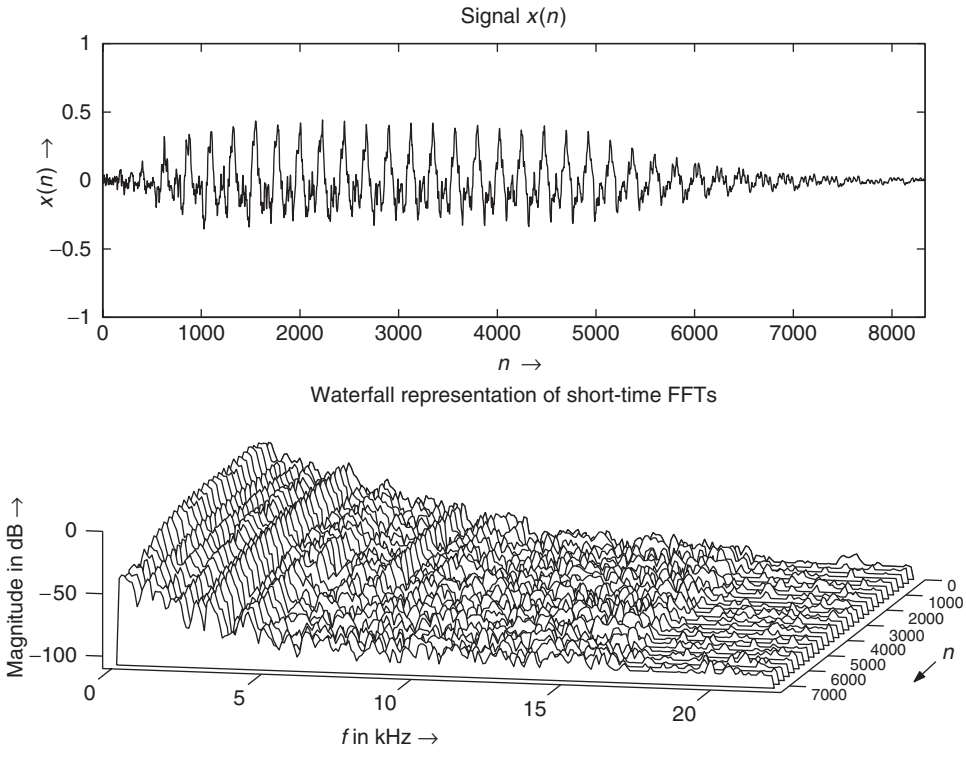

Figure 1.22 Waterfall representation via FFT of weighted segments.

1.3.3 Digital systems

A digital system is represented by an algorithm which uses the input signal $x(n)$ as a sequence (stream) of numbers and performs mathematical operations upon the input signal such as additions, multiplications and delay operations. The result of the algorithm is a sequence of numbers or the output signal $y(n)$. Systems which do not change their behavior over time and fulfill the superposition property [Orf96] are called linear time-invariant (LTI) systems. The input/output relations for a LTI digital system describe time-domain relations which are based on the following terms and definitions:

- Unit impulse, impulse response and discrete convolution;
- Algorithms and signal flow graphs.

For each of these definitions an equivalent description in the frequency domain exists, which will be introduced later.

Unit impulse, Impulse response and Discrete convolution

- Test signal: a very useful test signal for digital systems is the unit impulse

$$\delta(n) = \begin{cases} 1 & \text{for} \quad n = 0 \\ 0 & \text{for} \quad n \neq 0, \end{cases} \tag{1.7}$$

which is equal to one for $n = 0$ and zero elsewhere (see Figure 1.23).

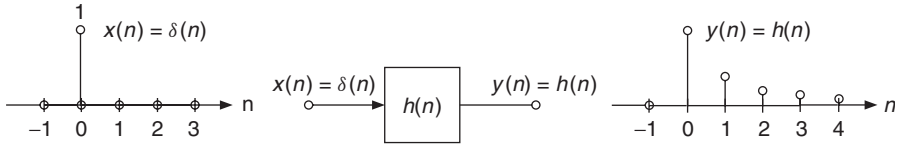

Figure 1.23 Impulse response $h(n)$ as a time-domain description of a digital system.

- Impulse response: if we apply a unit-impulse function to a digital LTI system, the digital system will lead to an output signal $y(n) = h(n)$, which is called the impulse response $h(n)$ of the digital system. The digital LTI system is completely described by the impulse response, which is pointed out by the label $h(n)$ inside the box, as shown in Figure 1.23.
- Discrete convolution: if we know the impulse response $h(n)$ of a digital system, we can calculate the output signal $y(n)$ from a freely chosen input signal $x(n)$ by the discrete convolution formula given by

$$y(n) = \sum_{k=-\infty}^{\infty} x(k) \cdot h(n-k) = x(n) * h(n), \tag{1.8}$$

which is often abbreviated by the second term $y(n) = x(n) * h(n)$. This discrete sum formula (1.8) represents an input–output relation for a digital system in the time domain. The computation of the convolution sum formula (1.8) can be achieved by the **MATLAB** function `y=conv(x,h)`.

Algorithms and signal flow graphs

The above given discrete convolution formula shows the mathematical operations which have to be performed to obtain the output signal $y(n)$ for a given input signal $x(n)$. In the following we will introduce a visual representation called a signal flow graph which represents the mathematical input/output relations in a graphical block diagram. We discuss some example algorithms to show that we only need three graphical representations for the multiplication of signals by coefficients, delay and summation of signals.

- A delay of the input signal by two sampling intervals is given by the algorithm

$$y(n) = x(n-2) \tag{1.9}$$

and is represented by the block diagram in Figure 1.24.

- A weighting of the input signal by a coefficient a is given by the algorithm

$$y(n) = a \cdot x(n) \tag{1.10}$$

and represented by a block diagram in Figure 1.25.

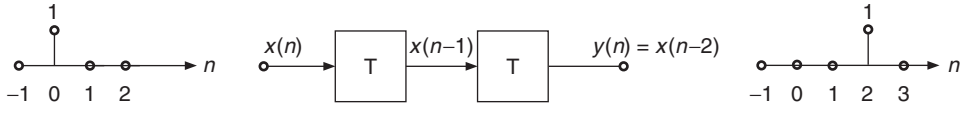

Figure 1.24 Delay of the input signal.

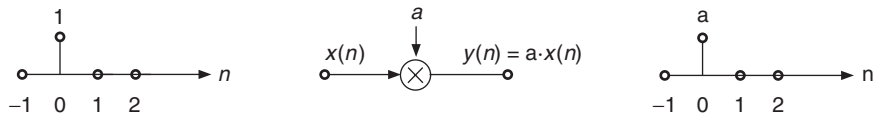

Figure 1.25 Weighting of the input signal.

- The addition of two input signals is given by the algorithm

$$y(n) = a_1 \cdot x_1(n) + a_2 \cdot x_2(n) \tag{1.11}$$

  and represented by a block diagram in Figure 1.26.

- The combination of the above algorithms leads to the weighted sum over several input samples, which is given by the algorithm

$$y(n) = \frac{1}{3}x(n) + \frac{1}{3}x(n-1) + \frac{1}{3}x(n-2) \tag{1.12}$$

  and represented by a block diagram in Fig. 1.27.

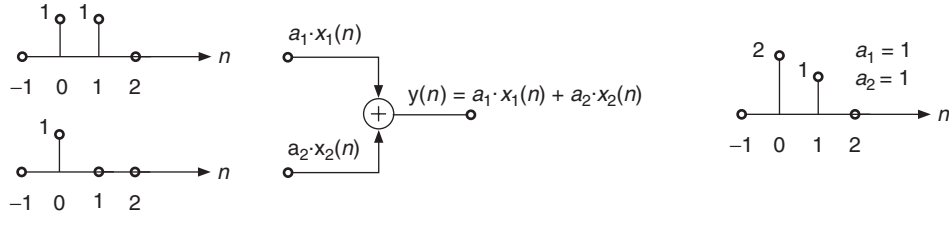

Figure 1.26 Addition of two signals $x_1(n)$ and $x_2(n)$.

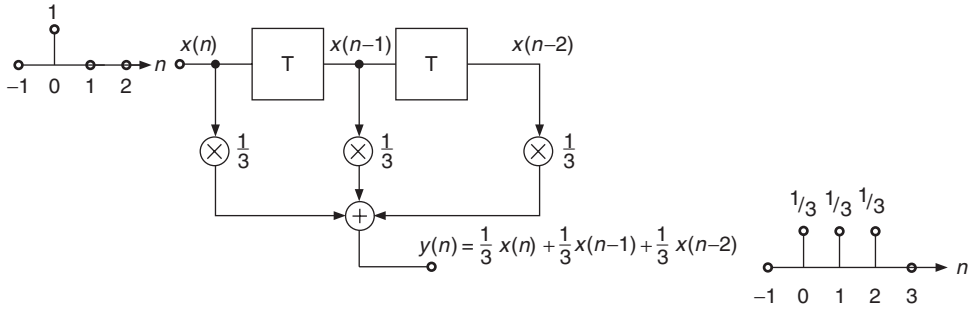

Figure 1.27 Simple digital system.

Transfer function and frequency response

So far our description of digital systems has been based on the time-domain relationship between the input and output signals. We noticed that the input and output signals and the impulse response of the digital system are given in the discrete time domain. In a similar way to the frequency-domain

description of digital signals by their spectra given in the previous subsection we can have a frequency-domain description of the digital system which is represented by the impulse response $h(n)$. The frequency-domain behavior of a digital system reflects its ability to pass, reject and enhance certain frequencies included in the input signal spectrum. The common terms for the frequency-domain behavior are the transfer function $H(z)$ and the frequency response $H(f)$ of the digital system. Both can be obtained by two mathematical transforms applied to the impulse response $h(n)$.

The first transform is the *Z-Transform*

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) \cdot z^{-n} \tag{1.13}$$

applied to the signal $x(n)$ and the second transform is the *discrete-time Fourier transform*

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n) \cdot e^{-j\omega n}, \tag{1.14}$$

$$\text{with} \quad \omega = 2\pi f / f_S \tag{1.15}$$

applied to the signal $x(n)$. Both are related by the substitution $z \leftrightarrow e^{j\omega}$. If we apply the Z-transform to the impulse response $h(n)$ of a digital system according to

$$H(z) = \sum_{n=-\infty}^{\infty} h(n) \cdot z^{-n} \tag{1.16}$$

we denote $H(z)$ as the *transfer function*. The transfer function is of special interest as it relates the Z-transforms of input signal and output signal of the described system by

$$Y(z) = H(z) \cdot X(z). \tag{1.17}$$

If we apply the discrete-time Fourier transform to the impulse response $h(n)$ we get

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h(n) \cdot e^{-j\omega n}. \tag{1.18}$$

Substituting (1.15) we define the *frequency response* of the digital system by

$$H(f) = \sum_{n=-\infty}^{\infty} h(n) \cdot e^{-j2\pi f / f_S n}. \tag{1.19}$$

Causal and stable systems

A realizable digital system has to fulfill the following two conditions:

- Causality: a discrete-time system is *causal*, if the output signal $y(n) = 0$ for $n < 0$ for a given input signal $x(n) = 0$ for $n < 0$. This means that the system cannot react to an input before the input is applied to the system.
- Stability: a digital system is stable if

$$\sum_{n=-\infty}^{\infty} |h(n)| < M_2 < \infty \tag{1.20}$$

holds. The sum over the absolute values of $h(n)$ has to be less than a fixed number $M_2 < \infty$.

The stability implies that the transfer function (Z-transform of impulse response) and the frequency response (discrete-time Fourier transform of impulse response) of a digital system are related by the substitution $z \leftrightarrow e^{j\omega}$. Realizable digital systems have to be *causal* and *stable* systems. Some Z-transforms and their discrete-time Fourier transforms of a signal $x(n)$ are given in Table 1.4.

Table 1.4 Z-transforms and discrete-time Fourier transforms of $x(n)$.

Signal	Z-transform	Discrete-time Fourier transform
$x(n)$	$X(z)$	$X(e^{j\omega})$
$x(n-M)$	$z^{-M} \cdot X(z)$	$e^{-j\omega M} \cdot X(e^{j\omega})$
$\delta(n)$	1	1
$\delta(n-M)$	z^{-M}	$e^{-j\omega M}$
$x(n) \cdot e^{j\omega_0 n}$	$X(e^{-j\omega_0} \cdot z)$	$X(e^{j(\omega-\omega_0)})$

IIR and FIR systems

IIR systems: A system with an infinite impulse response $h(n)$ is called an IIR system. From the block diagram in Figure 1.28 we can read the difference equation

$$y(n) = x(n) - a_1 y(n-1) - a_2 y(n-2). \tag{1.21}$$

The output signal $y(n)$ is fed back through delay elements and a weighted sum of these delayed outputs is summed up to the input signal $x(n)$. Such a feedback system is also called a recursive system. The Z-transform of (1.21) yields

$$Y(z) = X(z) - a_1 z^{-1} Y(z) - a_2 z^{-2} Y(z) \tag{1.22}$$

$$X(z) = Y(z)(1 + a_1 z^{-1} + a_2 z^{-2}) \tag{1.23}$$

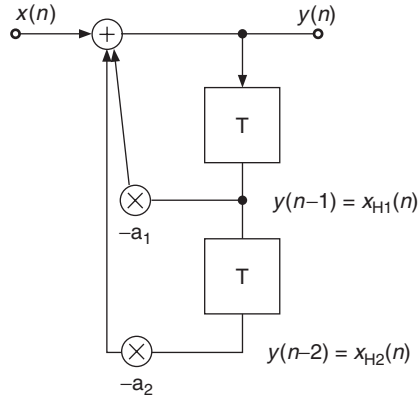

Figure 1.28 Simple IIR system with input signal $x(n)$ and output signal $y(n)$.

and solving for $Y(z)/X(z)$ gives transfer function

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2}}. \tag{1.24}$$

Figure 1.29 shows a special signal flow graph representation, where adders, multipliers and delay operators are replaced by weighted graphs.

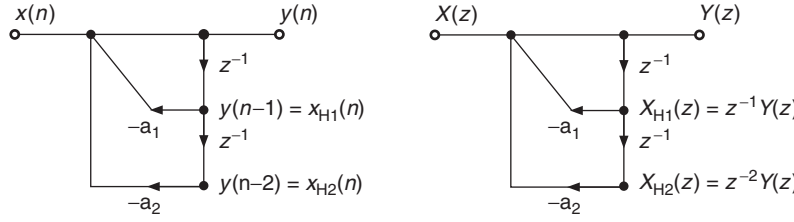

Figure 1.29 Signal flow graph of digital system in Figure 1.28 with time-domain description in the left block diagram and corresponding frequency-domain description with Z-transform.

If the input delay line is extended up to $N-1$ delay elements and the output delay line up to M delay elements according to Figure 1.30, we can write for the difference equation

$$y(n) = -\sum_{k=1}^{M} a_k \; y(n-k) + \sum_{k=0}^{N-1} b_k \; x(n-k), \tag{1.25}$$

the Z-transform of the difference equation

$$Y(z) = -\sum_{k=1}^{M} a_k \; z^{-k} Y(z) + \sum_{k=0}^{N-1} b_k \; z^{-k} X(z), \tag{1.26}$$

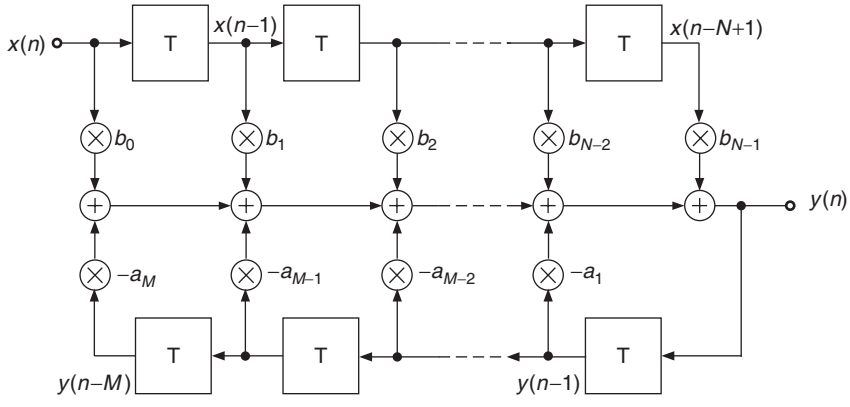

Figure 1.30 IIR system.

and the resulting transfer function

$$H(z) = \frac{\sum_{k=0}^{N-1} b_k z^{-k}}{1 + \sum_{k=1}^{M} a_k z^{-k}}. \tag{1.27}$$

The block processing approach for the IIR filter algorithm can be performed with the **MATLAB**/Octave function `y = filter(b, a, x)`, where `b` and `a` are vectors with the filter coefficients as above and `x` contains the input signal.

A sample-by-sample processing approach for a second-order IIR filter algorithm is demonstrated by M-file 1.8.

M-file 1.8 (`DirectForm01.m`)

```
% Author: U. Zölzer
% Impulse response of 2nd order IIR filter
% Sample-by-sample algorithm
clear

% Coefficients for a high-pass
a=[1, -1.28, 0.47];
b=[0.69, -1.38, 0.69];

% Initialization of state variables
xh1=0;xh2=0;
yh1=0;yh2=0;

% Input signal: unit impulse
N=20;  % length of input signal
x(N)=0;x(1)=1;

% Sample-by-sample algorithm
for n=1:N
y(n)=b(1)*x(n)+b(2)*xh1+b(3)*xh2 - a(2)*yh1 - a(3)*yh2;
xh2=xh1;xh1=x(n);
yh2=yh1;yh1=y(n);
end;

% Plot results
subplot(2,1,1)
stem(0:1:length(x)-1,x,'.');axis([-0.6 length(x)-1 -1.2 1.2]);
xlabel('n \rightarrow');ylabel('x(n) \rightarrow');
subplot(2,1,2)
stem(0:1:length(x)-1,y,'.');axis([-0.6 length(x)-1 -1.2 1.2]);
xlabel('n \rightarrow');ylabel('y(n) \rightarrow');
```

Computation of the frequency response based on the coefficients of the transfer function $H(z) = \frac{B(z)}{A(z)}$ can be made with the **MATLAB**/Octave function `freqz(b, a)`, while the poles and zeros can be determined with `zplane(b, a)`.

FIR systems: A system with a finite impulse response $h(n)$ is called an FIR system. From the block diagram in Figure 1.31 we can read the difference equation

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2). \tag{1.28}$$

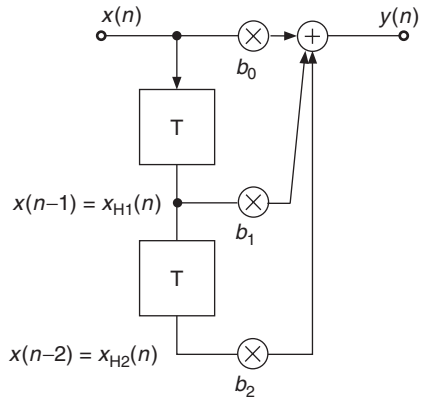

Figure 1.31 Simple FIR system with input signal $x(n)$ and output signal $y(n)$.

The input signal $x(n)$ is fed forward through delay elements and a weighted sum of these delayed inputs is summed up to the input signal $y(n)$. Such a feed-forward system is also called a non-recursive system. The Z-transform of (1.28) yields

$$Y(z) = b_0 X(z) + b_1 z^{-1} X(z) + b_2 z^{-2} X(z) \tag{1.29}$$

$$= X(z)(b_0 + b_1 z^{-1} + b_2 z^{-2}) \tag{1.30}$$

and solving for $Y(z)/X(z)$ gives transfer function

$$H(z) = \frac{Y(z)}{X(z)} = b_0 + b_1 z^{-1} + b_2 z^{-2}. \tag{1.31}$$

A general FIR system in Figure 1.32 consists of a feed-forward delay line with $N-1$ delay elements and has the difference equation

$$y(n) = \sum_{k=0}^{N-1} b_k \; x(n-k). \tag{1.32}$$

The finite impulse response is given by

$$h(n) = \sum_{k=0}^{N-1} b_k \; \delta(n-k), \tag{1.33}$$

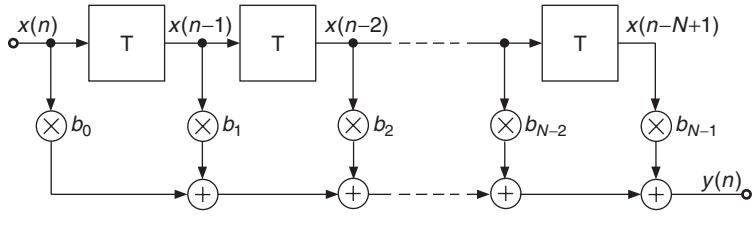

Figure 1.32 FIR system.

which shows that each impulse of $h(n)$ is represented by a weighted and shifted unit impulse. The Z-transform of the impulse response leads to the transfer function

$$H(z) = \sum_{k=0}^{N-1} b_k z^{-k}. \tag{1.34}$$

The time-domain algorithms for FIR systems are the same as those for IIR systems with the exception that the recursive part is missing. The previously introduced M-files for IIR systems can be used with the appropriate coefficients for FIR block processing or sample-by-sample processing.

The computation of the frequency response $H(f) = |H(f)| \cdot e^{j\angle H(f)}$ ($|H(f)|$ magnitude response, $\varphi = \angle H(f)$ phase response) from the Z-transform of an FIR impulse response according to (1.34) is shown in Figure 1.33 and is calculated by the following M-file 1.9.

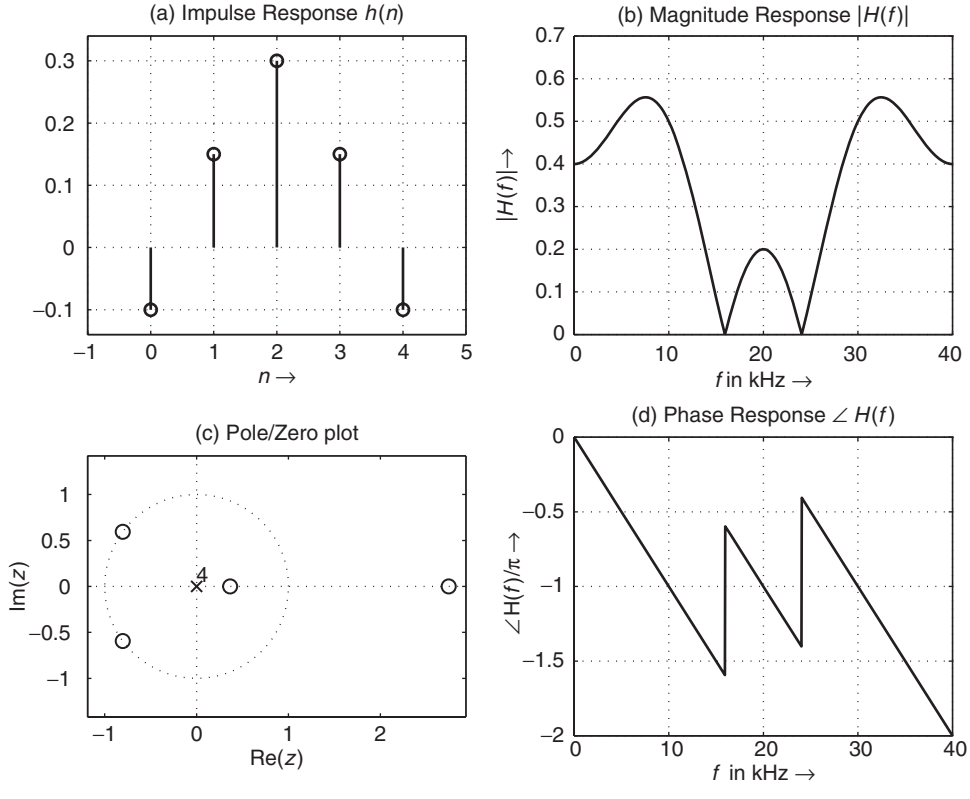

Figure 1.33 FIR system: (a) impulse response, (b) magnitude response, (c) pole/zero plot and (d) phase response (sampling frequency $f_S = 40$ kHz).

M-file 1.9 (`figure1_33.m`)

```
function magphasresponse(h)
% Author: U. Zölzer
FS=40000;
fosi=10;
```

```
if nargin==0
  h=[-.1 .15 .3 .15 -.1];
end
hmax=max(h);
hmin=min(h);
dh=hmax-hmin;
hmax=hmax+.1*dh;
hmin=hmin-.1*dh;

N=length(h);
% denominator polynomial:
a=zeros(1,N);
a(1)=1;

subplot(221)
stem(0:N-1,h)
axis([-1 N, hmin hmax])
title('a) Impulse Response h(n)','Fontsize',fosi);
xlabel('n \rightarrow','Fontsize',fosi)
grid on;

subplot(223)
zplane(h,a)
title('c) Pole/Zero plot','Fontsize',fosi);
xlabel('Re(z)','Fontsize',fosi)
ylabel('Im(z)','Fontsize',fosi)

subplot(222)
[H,F]=freqz(h,a,1024,'whole',FS);
plot(F/1000,abs(H))
xlabel('f in kHz  \rightarrow','Fontsize',fosi);
ylabel('|H(f)| \rightarrow','Fontsize',fosi);
title('b) Magnitude response |H(f)|','Fontsize',fosi);
grid on;

subplot(224)
plot(F/1000,unwrap(angle(H))/pi)
xlabel('f  in kHz \rightarrow','Fontsize',fosi)
ylabel('\angle H(f)/\pi \rightarrow','Fontsize',fosi)
title('d) Phase Response \angle H(f)','Fontsize',fosi);
grid on;
```

1.4 Conclusion

In this first chapter we introduced definitions and classifications of audio effects, to provide an overview of the territory to be explored. Then, some basic concepts of digital signals, their spectra and digital systems have been introduced. The description is intended for persons with little or no knowledge of digital signal processing. The inclusion of **MATLAB** M-files for all stages of processing may serve as a basis for further programming in the following chapters. As well as showing simple tools for graphical representations of digital audio signals we have calculated the spectrum of a signal $x(n)$ by the use of the FFT M-file

- `Xmagnitude=abs(fft(x))`
  `Xphase=angle(fft(x))`.

Time-domain processing for DAFX can be performed by block-based input–output computations which are based on the convolution formula (if the impulse response of a system is known) or difference equations (if the coefficients a and b are known). The computations can be done by the following M-files:

- ```
  y=conv(h,x)     %length of output signal l_y = l_h +l_x -1
  y=filter(b,a,x) %l_y = l_x
  ```

These M-files deliver an output vector containing the output signal $y(n)$ in a vector of corresponding length. Of course, these block processing algorithms perform their inner computations on a sample-by-sample basis. Therefore, we have also shown an example for the sample-by-sample programming technique, which can be modified according to different applications:

- ```
  y=dafxalgorithm(parameters,x)
  % Sample-by sample algorithm y(n)=function(parameters,x(n))
       for n=1:length(x),
       y(n)=....do something algorithm with x(n) and parameters;
       end;
  ```

That is all we need for DAFX exploration and programming, good luck!

References

[ABL+03] X. Amatriain, J. Bonada, A. Loscos, J. L. Arcos and V. Verfaille. Content-based transformations. *J. New Music Research*, 32(1): 95–114, 2003.

[AD98] D. Arfib and N. Delprat. Selective transformations of sound using time-frequency representations: An application to the vibrato modification. In *104th Conv. Audio Eng. Soc., Amsterdam*, 1998.

[AKZ02a] D. Arfib, F. Keiler and U. Zölzer. *DAFX–Digital Audio Effects*, First edition, Time-Frequency Processing, pp. 237–97. U. Zölzer ed., J. Wiley & Sons, Ltd, 2002.

[AKZ02b] D. Arfib, F. Keiler and U. Zölzer. *DAFX - Digital Audio Effects*, First edition, Source-filter processing, pp. 299–372. U. Zölzer ed., J. Wiley & Sons, Ltd, 2002.

[All77] J. B. Allen. Short term spectral analysis, synthesis and modification by discrete fourier transform. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 25(3): 235–8, 1977.

[ANS60] ANSI. *USA Standard Acoustic Terminology*. American National Standards Institute, 1960.

[AOPW99] S. Abrams, D. V. Oppenheim, D. Pazel and J. Wright. Higher-level composition control in lusic sketcher: Modifiers and smart harmony. In *Proc. Int. Computer Music Conf. (ICMC'99), Beijing*, pp. 13–6, 1999.

[AR77] J. B. Allen and L. R. Rabiner. A unified approach to short-time fourier analysis and synthesis. *Proc. IEEE*, 65(11): 1558–64, 1977.

[Arf98] D. Arfib. Different ways to write digital audio effects programs. In *Proc. DAFX-98 Digital Audio Effects Workshop*, pp. 188–191, Barcelona, November 1998.

[Arf99] D. Arfib. Visual representations for digital audio effects and their control. In *Proc. DAFX-99 Digital Audio Effects Workshop*, pp. 63–68, Trondheim, December 1999.

[Bar70] B. Bartlett. A scientific explanation of phasing (flanging). *J. Audio Eng. Soc*., 18(6): 674–5, 1970.

[BB96] T. Buzan and B. Buzan. *Mind Map Book*. Plume, 1996.

[BJ95] R. Bristow-Johnson. A detailed analysis of time-domain formant-corrected pitch-shifting algorithm. *J. Audio Eng. Soc*., 43(5): 340–52, 1995.

[Bla83] J. Blauert. *Spatial Hearing: the Psychophysics of Human Sound Localization*. MIT Press, 1983.

[Ble01] B. Blesser. An interdisciplinary synthesis of reverberation viewpoints. *J. Audio Eng. Soc*., 49(10): 867–903, 2001.

[BP89] J. C. Brown and M S. Puckette. Calculation of a narrowed autocorrelation function. *J. Ac. Soc. of America*, 85: 1595–601, 1989.

[Cab99] D. Cabrera. PsySound: a computer program for psychoacoustical analysis. In *Proc. Australian Ac. Soc. Conf., Melbourne*, pp. 47–53, November 1999.

[Cho71] J. Chowning. The simulation of moving sound sources. *J. Audio Eng. Soc*., 19(1): 1–6, 1971.

[CKC+04] P. Cano, M. Koppenberger, O. Celma, P. Herrera and V. Tarasov. Sound effects taxonomy management in production environments. In *Int. Conf. Audio Eng. Soc*., London UK, 2004.

[CPR95] A. Camurri, G. De Poli and D. Rocchesso. A taxonomy for sound and music computing. *Computer Music J*., 19(2): 4–5, 1995.

[Dat97] J. Dattoro. Effect design, part 2: Delay-line modulation and chorus. *J. Audio Eng. Soc*., pp. 764–88, 1997.

[dC04] A. de Cheveigné. *Pitch*, Pitch perception models. C. Plack and A. Oxenham eds, Springer-Verlag, Berlin, 2004.

[DH92] P. Desain and H. Honing. *Music, Mind and Machine: Studies in Computer Music, Music Cognition, and Artificial Intelligence*. Thesis Publishers, 1992.

[Dol86] M. Dolson. The phase vocoder: a tutorial. *Computer Music J*., 10(4): 14–27, 1986.

[DT96] S. Dubnov and N. Tishby. Testing for gaussianity and non linearity in the sustained portion of musical sounds. In *Proc. Journées Informatique Musicale (JIM'96)*, 1996.

[Dut91] P. Dutilleux. *Vers la machine à sculpter le son, modification en temps-réel des caractéristiques fréquentielles et temporelles des sons*. PhD thesis, University of Aix-Marseille II, 1991.

[Fav01] E. Favreau. Phase vocoder applications in GRM tools environment. In *Proc. of the COST-G6 Workshop on Digital Audio Effects (DAFx-01), Limerick*, pp. 134–7, 2001.

[Gav93] W. W. Gaver. What in the world do we hear? An ecological approach to auditory event perception. *Ecological Psychology*, 5(1): 1–29, 1993.

[Ger85] M. A. Gerzon. Ambisonics in multichannel broadcasting and video. *J. Audio Eng. Soc*., 33(11), 1985.

[GKP+05] C. Guastavino, B. F. Katz, J.-D. Polack, D. J. Levitin and D. Dubois. Ecological validity of soundscape reproduction. *Acta Acust. United Ac*., 91(2): 333–341, 2005.

[Har78] W. M. Hartmann. Flanging and phasers. *J. Audio Eng. Soc*., 26: 439–43, 1978.

[Hay96] S. Haykin. *Adaptive Filter Theory*, Third edition Prentice Hall, 1996.

[HMM04] D. Hargreaves, D. Miell and R. MacDonald. What do we mean by musical communication, and why it is important?, introduction of "*Musical communication (part 1)*" session, ICMPC CD-ROM. In *Proc. Int. Conf. Music Perc. and Cog*., 391–394, 2004.

[Hon95] H. Honing. The vibrato problem, comparing two solutions. *Computer Music J*., 19(3): 32–49, 1995.

[Lar98] J. Laroche. Time and pitch scale modification of audio signals. In M. Kahrs and K. Brandenburg, eds, *Applications of Digital Signal Processing to Audio & Acoustics*, pp. 279–309. Kluwer Academic Publishers, 1998.

[Lar01] J. Laroche. Estimating tempo, swing and beat locations in audio recordings. In *Proc. IEEE Workshop on Applications of Digital Signal Processing to Audio and Acoustics*, pp. 135–8, 2001.

[LD97] J. Laroche and M. Dolson. About this phasiness business. In *Proc. Int. Computer Music Conf. (ICMC'97), Thessaloniki*, pp. 55–8, 1997.

[LVKL96] T. I. Laakso, V. Välimäki, M. Karjalainen and U. K. Laine. Splitting the unit delay. In *IEEE Signal Proc. Mag*., pp. 30–60, 1996.

[MB90] R. C. Maher and J. Beauchamp. An investigation of vocal vibrato for synthesis. *Appl. Acoust*., 30: 219–45, 1990.

[MB96] P. Masri and A. Bateman. Improved modelling of attack transients in music analysis-resynthesis. In *Proc. Int. Computer Music Conf. (ICMC'96), Hong Kong*, pp. 100–3, 1996.

[MC90] E. Moulines and F. Charpentier. Pitch synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech Com*., 9(5/6): 453–67, 1990.

[ME93] C. Marvin and G. Ewers. *A Simple Approach to Digital Signal Processing*. Texas Instruments, 1993.

[Mit01] S.K Mitra. *Digital Signal Processing–A Computer-Based Approach*, Second edition. McGraw-Hill, 2001.

[MK73] M. Mathews and J. Kohut. Electronic simulation of violin resonances. *J. Ac. Soc. of America*, 53(6): 1620–6, 1973.

[ML95] E. Moulines and J. Laroche. Non-parametric technique for pitch-scale and time-scale modification. *Speech Com*., 16: 175–205, 1995.

[MM01] A. Marui and W. L. Martens. Perceptual and semantic scaling for user-centered control over distortion-based guitar effects. In *110th Conv. Audio Eng. Soc., Paris, France*, 2001. Preprint 5387.

[Mol75] J. Molino. Fait musical et sémiologie de la musique. *Musique en Jeu*, 17: 37–62, 1975.

[Moo79] J. A. Moorer. About this reverberation business. *Computer Music J*., 3(2): 13–8, 1979.

[Moo90] F. R. Moore. *Elements of Computer Music*. University of California, San Diego, Prentice Hall Inc., 1990.

[MQ86] R. J. McAulay and T. F. Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Trans. Acoust., Speech Signal Proc*., 34(4): 744–54, 1986.

[MSY98] J. McClellan, R. Schafer and M. Yoher. *DSP FIRST: A Multimedia Approach*. Prentice-Hall, 1998.

[MWdSK95] S. McAdams, S. Winsberg, G. de Soete and J. Krimphoff. Perceptual scaling of synthesized musical timbres: common dimensions, specificities, and latent subject classes. *Psychol. Res*., 58: 177–92, 1995.

[Nat75] J.-J. Nattiez. *Fondements d'une Sémiologie de la Musique*. U. G. E., Coll. 10/18, Paris, 1975.

[Orf96] S.J. Orfanidis. *Introduction to Signal Processing*. Prentice-Hall, 1996.

[Por76] M. Portnoff. Implementation of the digital phase vocoder using the fast Fourier transform. *IEEE Trans. Acoust. Speech Signal Proc*., 24(3): 243–8, 1976.

[PPPR96] G. De Poli, A. Picialli, S. T. Pope, and C. Roads (eds). *Musical Signal Processing*. Swets & Zeitlinger, 1996.

[Puc95] M. S. Puckette. Phase-locked vocoder. In *Proc. IEEE ASSP Conf. on Appl. of Signal Proc. Audio Acoust. (Mohonk, NY)*, 1995.

[Pul97] V. Pulkki. Virtual sound source positioning using vector base amplitude panning. *J. Audio Eng. Soc*., 45(6): 456–66, 1997.

[Rab] C. A. Rabassó. L'improvisation: du langage musical au langage littéraire. *Intemporel: Bulletin de la Société Nationale de Musique*, 15.

[RDS+99] S. Rossignol, P. Depalle, J. Soumagne, X. Rodet and J.-L. Collette. Vibrato: Detection, estimation, extraction, modification. In *Proc. COST-G6 Workshop on Digital Audio Effects (DAFx-99), Trondheim*, 1999.

[Roa96] C. Roads. *The Computer Music Tutorial*. MIT Press, 1996.

[RW99] J.-C. Risset and D. L. Wessel. *Exploration of Timbre by Analysis and Synthesis*, pp. 113–69. D. Deutsch, Academic Press, 1999.

[Sch66] P. Schaeffer. *Traité des Objets Musicaux*. Seuil, 1966.

[Sch77] R. M. Schafer. *The Tuning of the World*. Knopf: New York, 1977.

[Sea36] C. E. Seashore. Psychology of the vibrato in voice and speech. *Studies Psychol. Music*, 3, 1936.

[She82] R. Shepard. Geometrical approximations to the structure of musical pitch. *Psychol. Rev*., 89(4): 305–33, 1982.

[SL61] M. R. Schrœder and B. Logan. "Colorless" artificial reverberation. *J. Audio Eng. Soc*., 9: 192–7, 1961.

[Sla85] W. Slawson. *Sound Color*. University of California Press, 1985.

[Smi84] J. O. Smith. An allpass approach to digital phasing and flanging. In *Proc. Int. Computer Music Conf. (ICMC'84), Paris*, pp. 103–8, 1984.

[SS90] X. Serra and J. O. Smith. A sound decomposition system based on a deterministic plus residual model. *J. Ac. Soc. of America, Sup. 1*, 89(1): 425–34, 1990.

[SSAB02] J. O. Smith, S. Serafin, J. Abel and D. Berners. Doppler simulation and the Leslie. In *Proc. Int. Conf. on Digital Audio Effects (DAFx-02), Hamburg*, pp. 13–20, 2002.

[Sun87] J. Sundberg. *The Science of the Singing Voice*. Northern Illinois University Press, 1987.

[VGD05] V. Verfaille, C. Guastavino and P. Depalle. Perceptual evaluation of vibrato models. In *Colloq. Interdisc. Musicol., Montréal (CIM'05)*, 2005.

[VGT06] Vincent Verfaille, Catherine Guastavino and Caroline Traube. An interdisciplinary approach to audio effect classification. In *Proc. 9th Int. Conf. Digital Audio Effects (DAFx-06), Montreal, Canada*, pp. 107–13, 2006.

[VLM97] T. Verma, S. Levine and T. Meng. Transient modeling synthesis: a flexible analysis/synthesis tool for transient signals. In *Proc. Int. Computer Music Conf. (ICMC'97), Thessaloniki*, 164–167, 1997.

[VWD06] V. Verfaille, M. M. Wanderley and Ph. Depalle. Mapping strategies for gestural control of adaptive digital audio effects. *J. New Music Res*., 35(1): 71–93, 2006.

[VZA06] Vincent Verfaille, U. Zölzer and Daniel Arfib. Adaptive digital audio effects (A-DAFx): A new class of sound transformations. *IEEE Trans. Audio, Speech and Lang. Proc*., 14(5): 1817–1831, 2006.

[WM01] O. Warusfel and N. Misdariis. Directivity synthesis with a 3D array of loudspeakers - Application for stage performance. In *Proc. of the COST-G6 Workshop on Digital Audio Effects (DAFx-01), Limerick*, pp. 232–6, 2001.

[Zöl02] U. Zölzer (ed). *DAFX–Digital Audio Effects*, First edition. J. Wiley & Sons, Ltd, 2002.
[Zöl05] U. Zölzer. *Digital Audio Signal Processing*, Second edition. John Wiley & Sons, Ltd, 2005.
[ZS65] E. Zwicker and B. Scharf. A model of loudness summation. *Psychol. Rev*., 72: 3–26, 1965.
[Zwi77] E. Zwicker. Procedure for calculating loudness of temporally variable sounds. *J. Ac. Soc. of America*, 62(3): 675–82, 1977.