



Acknowledgements

Lemur is based on `mqa`, a tool developed by Robert Maher and James Beauchamp at the University of Illinois.

Macintosh gurus—Kurt Hebel, John Brewer, Dan Walkowski

Sinusoidal modeling guru—Lippold Haken

Overview

Sinusoidal modeling

Lemur is based on the technique of sinusoidal modeling. The premise of sinusoidal modeling is that any signal can be represented as a summation of sine waves of various amplitudes, frequencies, and starting phases. In the past, most implementations of sinusoidal modeling of nonperiodic signals have performed a series of Short Time Fourier transforms (STFTs) to determine the amplitudes, frequencies, and phases of the sinusoidal components for small segments of the signal. These amplitudes, frequencies,

Lemur Documentation p. 2

and phases are then used to synthesize a small segment of the signal, and the small segments are strung together to construct the complete, synthesized signal. This straightforward sinusoidal implementation has several problems. The STFT does not report exact frequencies for peaks in the frequency spectrum of the signal. An N-point STFT reports frequency components only at multiples of $\text{samplingFrequency} / N$, limiting the usefulness of this model for frequency-scale modification of a signal. Phase uncertainty and discontinuities can create audible artifacts when the synthesis segments are strung together, limiting the usefulness of this model for time-scale modification of a signal. (A more detailed description of the STFT may be found in signal processing texts.) Pitch-tracking methods solve some of these problems, but their use is restricted to the class of monophonic, strongly harmonic signals (Grey 1975, Haken 1989).

McAulay-Quatieri technique

In a 1985 technical report from M. I. T. Lincoln Labs, McAulay and Quatieri proposed a sinusoidal analysis technique for speech processing. The premise of the MQ technique is that a sound can be represented by a collection of sinusoidal components (called tracks), each with time-varying amplitude and frequency. To construct these tracks, STFTs are performed on a signal at regular intervals, called frames. Amplitude peaks in the resulting frequency spectra are identified, and parabolic interpolation is used to obtain a close approximation of the exact spectral peak frequencies. These peaks are the most prominent frequencies in the sound at that instant. The peaks in adjacent frames are compared, and peaks of similar frequencies are matched. A continuous chain of these matched peaks is a track. To ensure smooth tracks, the MQ analysis minimizes the difference between the frequencies of the peaks being matched. A peak that is not matched represents the birth or death of a track. MQ synthesis uses cubic phase interpolation to reduce phase uncertainty and eliminate phase discontinuities (McAulay and Quatieri 1985).

Lemur

Lemur uses the MQ technique of creating sinusoidal tracks by linking interpolated peaks in the short time spectra of a signal, and performing cubic phase interpolation (also called “phase unwrapping”) during synthesis. Lemur provides some extensions to the basic McAulay-Quatieri technique.

Frequency Bins

The original MQ technique attempted to model psychoacoustic masking effects by suggesting that an amplitude threshold for peak detection should be based on the loudest peak in each frame. In other words, when the sound is loud, only loud peaks need to be represented, since quieter ones will be masked. When the sound is quiet, the quieter peaks are much more important. Unfortunately, this global threshold ignores the importance of frequency in masking effects. For example, a high frequency rarely masks a low one (Haken 1989, Sciarabba 1991). Lemur provides a refinement of the original MQ amplitude threshold by breaking the frequency domain into logarithmically sized bins. The loudest peak in each bin is determined, and an amplitude threshold for each bin is based on its loudest peak. This allows quiet peaks to be ignored in a bin containing loud peaks, while detecting quiet peaks in a bin without loud peaks. This is not an accurate model of the effect of frequency in masking, but is an approximation which presents a significant improvement over the original model. A psychoacoustically accurate model is computationally prohibitive, and may not yield perceptibly

Lemur Documentation p. 4
more accurate syntheses.

Hysteresis

In examining the results of an MQ analysis, one often observes a track which dies out and another track which is born a few frames later at roughly the same frequency. These are best understood as two portions of the same track. Earlier attempts (Serra, 1989) to facilitate this representation allowed tracks to lie dormant for a specified number of frames before dying out. A dormant track had zero amplitude, but participated in peak matching. The dormancy representation gave a more intuitive and visually pleasing graph of the analysis, but did nothing to reduce the audible effects of low amplitude tracks repeatedly dying and being reborn (this effect has been affectionately called the “doodley-doo” effect). Lemur reduces this effect by allowing the specification of a track amplitude hysteresis. This is the amount by which a track may dip below the amplitude threshold while still participating in synthesis. Hysteresis differs from dormancy in that the tracks in the

hysteresis range are synthesized at the amplitudes reported from the frequency spectra, rather than at zero amplitude.

Parameters

Lemur has many parameters which may be adjusted to produce a good analysis (i.e., an analysis which yields good syntheses). Lemur initializes these parameters at startup to values which seem to work reasonably well under a variety of conditions. The optimality of the parameter settings depends heavily on the samples file being analyzed, the available storage, and the patience of the user. The user is therefore encouraged to liberally adjust the parameters from their default values, and to try several different sets of parameters to find the best settings for a particular analysis. An explanation of Lemur's parameter set is included below. At any time, the parameters may be reset to their initial values using the RESET PARAMETERS menu option. All Lemur Parameters are persistent. That is, all Lemur parameters are remembered from one analysis or synthesis to the next. The parameters are reset only when RESET PARAMETERS is selected from the Edit menu, or when the application is closed.

Analysis

During analysis, Lemur analyzes an AIFF samples file and creates a Lemur analysis file. While the analysis is in progress, the Lemur Graph window displays the tracks as they are created. The smaller window gives you an idea of how long you will have to wait.

Analysis Window

During analysis, Lemur performs spectral analysis on successive overlapping blocks of samples using a fast Fourier transform (FFT). The input to each FFT is a windowed set of samples from the input file. A window is used to reduce the spectral artifacts introduced by analyzing a segment of samples from a longer signal. The choice of window function is a trade-off between time and frequency resolution. This trade-off is governed

Lemur Documentation p. 6

by the width of the main lobe of the window spectrum and the height of the side lobes relative to the height of the main lobe (for a discussion of the use of window functions in spectral analysis, see Oppenheim and Schaffer 1989, Maher 1989, Serra 1989, Fitz 1992). Lemur uses the Kaiser window (Kaiser 1974, Kaiser and Schaffer 1980), which has a shaping parameter to control the balance between main lobe width and side lobe magnitude. The sidelobe level and main lobe width (in Hertz) may be adjusted using the ANALYSIS WINDOW menu option. Lemur computes the shape and length of the analysis window from these parameters.

Lemur can (and will by default) automatically compute an appropriate sidelobe level and main lobe width. If the COMPUTE WINDOW PARAMETERS AUTOMATICALLY option is selected, Lemur will set the sidelobe level to be the sum of the Peak Amplitude Range and Track Amplitude Hysteresis (see the discussion of THRESHOLDS, below). The main lobe width will be chosen to give a window length of 511 samples. The difference between the FFT length (1024 samples) and the window length is made up by zero padding (filling out the remainder of the FFT with zeros). This results in spectral interpolation, which makes the peaks easier to find. Best results are obtained by zero padding by at least a factor of two (i.e. the FFT is at least twice as long as the window) (Oppenheim and Schaffer 1989). Thus, Lemur will choose a window of length 511 samples if automatic window computation is selected.

Frame Length

During analysis, Lemur computes frequency spectra of overlapping blocks of samples. The time between successive spectra may be set using the FRAME LENGTH menu option. A frame contains the prominent frequency components selected from a single frequency spectrum during analysis. During synthesis, a number of samples corresponding to the Frame Length is generated for each pair of consecutive frames. The Frame Length is generally shorter than the FFT length, so that each sample is involved in several FFT's. This is especially important when you intend to time stretch during synthesis. Considerable overlap yields the best results, but also longer analysis times and larger analysis files. The default Frame Length is 3 milliseconds, which is about 66 samples at a sample rate of 22050 kHz.

Thresholds

Lemur uses a system of three thresholds to select the prominent frequency components from the frequency spectra:

TRACK BIRTH THRESHOLD • In addition to the time-varying threshold for peak detection presented in the original MQ algorithm (refer to the discussion of Frequency Bins above), Lemur imposes an absolute birth threshold. This constant threshold is meant to represent the noise floor of the signal. The quietest allowable birth is always determined by the maximum of the two thresholds. Tracks may have amplitude lower than this threshold, due to hysteresis, but a track will never be born at an amplitude lower than this threshold. The default threshold is -60 dB.

PEAK AMPLITUDE RANGE • Lemur locates the loudest peak in each frequency bin and subtracts the Peak Amplitude Range from its amplitude to determine the peak detection threshold. Reducing the range generally means an overall reduction in the number of peaks, but can have a detrimental effect on the quality of the synthesized sound. The default range is 20 dB.

TRACK AMPLITUDE HYSTERESIS • This is the amount by which a peak corresponding to a track continuation may fall below the time-varying threshold computed from the Birth Threshold and Range. Peaks are detected in this range, but if they fail to match to an existing track, they are discarded. The default hysteresis is 0 dB (no hysteresis). Hysteresis of 5 or 10 dB often improves the quality of the analysis, especially for non-speech signals, and may allow the Peak Amplitude Range to be reduced.

Frequency Drift

Lemur imposes a neighborhood for peak matching which varies with

Lemur Documentation p. 8

frequency. FREQUENCY DRIFT is the amount by which the frequency of a track will be allowed to vary over the duration of a frame, and is expressed as a percentage of the matching frequency (i.e. when matching to a 500 Hz peak, with a Frequency Drift of 1.0%, only peaks with frequencies between 495 Hz and 505 Hz will be considered for track formation). The default Frequency Drift of 2.0 percent seems to work well for most signals.

Pre-Filtering

Lemur can apply an adaptive filter to the analysis input signal. The adaptive filter will work to remove noise components from the signal before the analysis is performed. When this option is used, the Lemur analysis produces a Pre-Filter file, which is an AIFF samples file containing the noise extracted by the pre-filtering algorithm, in addition to the Lemur analysis file.

The USE PRE-FILTERING option in Lemur is part of ongoing research at the CERL Sound Group at the University of Illinois, and is probably not very useful in its current state. By default, Lemur does not apply the Pre-Filtering algorithm. Users wanting to experiment with the noise filter are encouraged to consult an adaptive filtering text (Haykin 1991) for an explanation of the parameters associated with the adaptive filter.

Synthesis

During Synthesis, Lemur examines an MQ file and produces a new AIFF samples file. While the synthesis is running, the Lemur Graph displays the tracks as they are read from the MQ file. Some parameters of the synthesis file may be specified from the Synthesis menu. By default, Lemur will create a synthesis file using 16-bit samples, at the sample rate of the AIFF file from which the Lemur file was produced. Lemur can also create 8-bit samples files, if desired, for use with other Macintosh applications (the Macintosh has 8-bit D/A converters) using the SAMPLE SIZE option (however, the synthesis file will still be an AIFF file, not an SND resource). The sample rate of the synthesis file may be set independently of the original file using the SAMPLE RATE

option.

The **AMPLITUDE SCALING** menu option allows the amplitude of the synthesized signal to be scaled up or down from the amplitude of the analyzed signal. If the amplitude scale factor is set to 1.0, the amplitude of the synthesized signal will be (approximately) equal to that of the analyzed signal (provided a sufficient number of tracks are obtained in the analysis). Scale values greater than one cause the amplitude of the synthesized signal to be greater than that of the analyzed signal. Scale values greater than one should be used with caution, as distortion may result.

Users of Lemur™ 2.0 may need to use scale values of less than 1.0 if they wish to synthesize from their old analysis files. Due to changes in internal scaling, analysis files from earlier versions of Lemur will distort very badly if the synthesis is not scaled down. For Lemur™ 2.0 analysis files, setting the amplitude scale factor to 0.01

will produce a synthesis of amplitude (approximately) equal to that of the original signal.

Lemur no longer produces normalized syntheses (syntheses whose largest sample is 1.0, or full amplitude) by default, as did versions 2.0 and earlier. Synthesis time has been reduced by half by eliminating the normalization pass. The current default synthesis has amplitude (approximately) equal to that of the analyzed signal. If a normalized synthesis is desired, the user may select the `AMPLITUDE SCALING` menu option, and choose `AUTO-NORMALIZE SYNTHESIS`. When this option is selected, Lemur synthesizes in two passes and produces full-amplitude syntheses.

If the `PLAY AIFFS WHEN DONE` option is selected, Lemur will play the samples file it has created upon completing the synthesis of an AIFF (samples) file.

Lemur can be used to synthesize control files for the SOS (Sum Of Sines) algorithm

designed by Lippold Haken at the University of Illinois CERL Sound Group for use with the Kyma System. Kyma System owners may contact the author for information about the SOS algorithm and its use. The GENERATE SOS FILE option will not be useful to anyone lacking a Kyma System, and the GENERATE AIFF FILE option should be checked instead. By default, Lemur generates AIFF samples files during synthesis. The GENERATE SOS FILE option is part of ongoing research at the CERL Sound Group and its use is not encouraged at this time.

Modification

Lemur allows time- and frequency-scale modification of an analyzed signal using the options in the Modify menu.

Scaling and Shifting

One of the most important features of sinusoidal modeling techniques is the ability to perform various transformations which are difficult in the time domain. Lemur provides three transformations during synthesis.

TIME SCALING is performed by adjusting the temporal spacing between frames. Unlike playing your 33 RPM records at 45 RPM, Time Scaling changes the duration of your sound without changing its pitch. If lengthening your sound produces strange frequency artifacts, try reanalyzing with a smaller hop size.

FREQUENCY SCALING multiplies every frequency in the MQ file by a given value before synthesizing. This preserves frequency ratios, but does not model the fixed formants of speech. As a result, frequency-scaled speech will not sound like the original speaker talking at a higher pitch. Note that frequency scaling does not change the duration of your sound. If you scale up your frequencies too far, some of them will fall above the Nyquist rate and may cause aliasing. Although Lemur does make an effort to avoid synthesizing tracks that will alias, it will not catch all of them.

FREQUENCY SHIFTING adds a given value to every frequency in the MQ file before synthesizing. This does not preserve frequency ratios, but (for small shifting values) tends to preserve fixed formant structures. If both frequency scaling and frequency shifting are used in synthesis, the scaling is performed first (e.g. if a frequency scale of 2.0 and a frequency shift of 50 Hz are used, then a 100 Hz peak will be transformed to 250 Hz).

Control Files

The three transformations discussed above can be controlled in two different ways—by specifying constant values or by specifying a control file. When a control file is specified, Lemur reads a new value from the control file for each frame. Thus, the control file provides time-varying control over shifting and scaling. You can construct these control files (which are ordinary AIFF samples files) using whatever tools you have, or with Lemur's simple

Lemur Documentation p. 14

breakpoint editor, which can generate files with step functions or linear segments. A limit on the amount of scaling or shifting affected by a control file is set using the CONTROL DEFAULTS option. The limits may be set independently. These limits effect scaling and shifting only from control files, and impose no limit on the specification of constant values for scaling or shifting.

Control files are interpreted in several ways, depending on what they are used to control. When a control file is used for time scaling, the samples in the control file are normalized to values between +1 and -1. The time scale is the Maximum Time Scale From File (specified in CONTROL DEFAULTS) raised to this power. Thus, if the Maximum Time Scale is set to ten (10) you can expand or contract by a factor of ten. A control file sample value of zero has no effect. The same algorithm is used for frequency scaling, thus you can multiply or divide your frequencies by a factor of ten, when the Maximum Frequency Scale is set to ten. Again, a sample value of zero has no effect. When a control file is used for frequency shifting, the samples in the control file are normalized to plus or minus the Maximum Frequency Shift. As

usual, a sample value of zero has no effect. The default maximum scales values are 10.0, and the default maximum frequency shift is 11025 Hz.

The control files facility should prove useful to composers seeking novel effects. Time-varying time scaling is also useful for research in speech processing. Constant time scaling does not model what happens when people speak slowly, since the consonants are stretched along with the vowels. If you construct an algorithm which examines the Lemur file and decides which parts are vowels and which parts are consonants, you can automatically construct a control file which will stretch the vowels while leaving the consonants untouched. Similar ideas might be used to stretch the sustain of a violin tone while leaving the attack transient untouched.

Control files may be generated using the MAKE CONTROL FILE menu option. You may use the breakpoint editor to specify a series of up to ten time and amplitude points for the control file. The time points are of only relative significance. Lemur reads the control file at the correct relative position for each frame, so the absolute duration of the control file is unimportant. Control files generated by Lemur are 8 bit, 1 kHz, AIFF files, and any single-channel AIFF may be used as a control file.

MQ file format

The MQ file produced by a Lemur analysis begins with a header which contains the following information:

2 bytes	signed integer	file format number
2 bytes	signed integer	header length
4 bytes	floating point	birth threshold
4 bytes	floating point	peak amplitude range
4 bytes	floating point	track amplitude hysteresis
4 bytes	floating point	main lobe width of analysis window
4 bytes	floating point	sidelobe level of analysis window
4 bytes	floating point	analysis frame length
4 bytes	signed integer	original number of samples
4 bytes	floating point	analysis sample rate
4 bytes	floating point	frequency drift
2 bytes	signed integer	length of pre-filter used during analysis

Following this header, the file contains successive frames in chronological order. At

the beginning of each frame is a two-byte integer which specifies how many peaks are in that frame. Empty frames are allowed. The peaks are presented in order from lowest frequency to highest. Each peak has the following format:

4 bytes	floating point	magnitude
4 bytes	floating point	frequency
4 bytes	floating point	interpolated frequency / starting phase of track
2 bytes	integer	next peak index
2 bytes	integer	label

Lemur retains phase information only for the first peak in a track. For successive peaks in a track, Lemur performs cubic phase interpolation and stores the interpolated frequency at the midpoint between two peaks. During synthesis, Lemur performs parabolic frequency interpolation for each track using the peak frequencies and the interpolated frequency. Since the frequency and phase are related by an integral, parabolic frequency interpolation is equivalent to cubic phase interpolation. Lemur uses the former because it is faster to compute and because it makes the analysis files easier to interpret and reduces the computation needed to perform modified (scaled) syntheses. A peak's interpolated frequency is the track frequency at a point halfway between that peak and the previous peak in the track. For the first peak in a track, this field contains the starting phase for the track.

The next peak index is used to connect a

peak with the next peak in its track by giving the ordinality of a peak in the next frame. For example, if a given peak has the value 5 for its next peak index, then that peak should be connected with the fifth peak in the next frame.

The label field is currently unused in Lemur, and is reserved for future use by the LemurEdit program, written by Bryan Holloway at the University of Illinois CERL Sound Group. Users who store data in this field do so at their own risk.

The file format number for Lemur 3.0 is 2185. This number may change for future versions of Lemur. Lemur analysis files are of type MQAN and their creator is LEMR.

File Information

Information about Lemur analysis files and AIFF samples files may be obtained using the GET FILE INFO menu option. File size and creation date are given for both types of

files. Analysis parameters are given for Lemur analysis files, along with information about the original samples file. Sample size and rate and file duration are given for AIFF samples files.

Lemur automatically updates analysis files which use file formats from earlier Lemur releases. Whenever SYNTHESIS or GET FILE INFO is attempted on an old analysis file, Lemur will offer to update the file. Old files may be updated without using SYNTHESIS or GET FILE INFO by using the UPDATE LEMUR FILE menu option.

Lemur can play an AIFF samples file using the PLAY AIFF FILE menu option. This option can be used to audition sounds for analysis, to hear the results of a synthesis, or to compare a synthesis with the original signal. Lemur can play any AIFF file, created by any program. The only restrictions on the file to be played are that it must be short enough to fit in the memory allotted to Lemur, and the sample rate must

be low enough that your Macintosh can keep up (up to 22257 Hz seems to work fine on most machines). The samples are converted to eight bits and played

back through the Macintosh speaker. While this does not provide sufficient fidelity for careful scrutiny of a synthesis, it is adequate for getting a rough idea about the quality of a synthesis, or the effect of some synthesis modification.

Lemur Display

While an analysis or synthesis is in progress, the Lemur Graph window displays the tracks as they are generated (during an analysis) or synthesized (during a synthesis).

The Display menu allows the user to turn off the track display, if desired, or to display a spectrogram of an analysis in progress.

DISPLAY SPECTROGRAM works only on color or gray-scale monitors. If the monitor is in Black and White mode, DISPLAY SPECTROGRAM will be disabled. DISPLAY SPECTROGRAM works only during an analysis, because, of course, during a synthesis, the track display is itself an accurate spectrogram of the synthesis file.

Lemur may run slightly faster when both displays are turned off. Lemur will run very much slower when the spectrogram display is turned on.

The Lemur Graph window may be closed at any time by clicking in its close box, as with any other Macintosh window. It may be recalled by selecting either of the options in the Display menu.

Scheduling

The Schedule menu lists the jobs Lemur is working on. The jobs are identified by the name of the file being created. The current job is in bold print at the top of the Schedule menu. Selecting items on the Schedule menu has no effect.

Multiple tasks may be scheduled in Lemur. While a process is running, another may be initiated. The jobs are queued up, placed on the Schedule, and processed in the order that they were scheduled. Processes always use the parameter settings that are in place at the

time the process was scheduled, so when one process is running, the parameters may be changed for a new process, and when the new process has been scheduled, they can be changed again for yet another process, etc. The current process can be halted using the HALT PROCESS option on the File menu. Processes depending on a halted process (i.e. needing files that were to be generated by the halted process) will also be voided, although they will not be removed from the schedule until they reach the top. The HALT ALL PROCESSES option removes all processes from the queue.

Known Problems

Lemur may behave strangely if the program is run from a volume mounted with AppleShare or if you attempt to analyze a file on a remote volume. Moving the Lemur Graph window between monitors is not recommended.

Lemur has been reported to crash a Mac SE30 with a small hard disk and little RAM,

especially when synthesizing at very low sample rates.

When Lemur crashes, it tries to delete any incomplete files that it generated from your disk. In some cases, it will be unable to do so, and a bogus file may be left lying around. In particular, if Lemur fills up your disk, it will be unable to delete any of the offending files.

References

Robert Crawford Maher, *An Approach for the separation of voices in composite musical signals*. Ph. D. dissertation, 1989, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign.

John Grey, *An Exploration of Musical Timbre*. Department of Music Report No. STAN-M-2, Ph.D. dissertation, 1975, Department of Physiology, Stanford University.

Kelly Fitz, *Time and Frequency Scale Modification of Audio Signals Using an Extended Sinusoidal Model*. M. S. thesis, 1992, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign.

Lippold Haken, *Real-time Fourier Synthesis of Ensembles with Timbral Interpolation*. Ph. D. dissertation, 1989, Department of Electrical and Computer Engineering,

Lemur Documentation p. 26

University of Illinois at Urbana-Champaign.

Simon Haykin, *Adaptive Filter Theory*.

Englewood Cliffs, NJ, Prentice Hall, 1991.

J. F. Kaiser, *Nonrecursive Digital Filter*

Design Using the $I_{0-\sinh}$ Window Function. IEEE Int. Symp. on Circuits and Syst, Apr. 1974, pp. 20-23.

J. F. Kaiser and R. W. Schafer, *On the Use of the $I_{0-\sinh}$ Window for Spectrum Analysis*. IEEE Trans. on Acoustics, Speech, and Signal Processing, vol. 28, no. 1, pp. 105-107.

A. V. Oppenheim and R. W. Schafer, *Discrete Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

T. F. Quatieri and R. J. McAulay, *Speech Analysis/Synthesis Based on a Sinusoidal Representation*. Technical Report 693, Lincoln Laboratory, M. I. T.

John Sciarabba, *Psychoacoustics in Sound Synthesis*. M. S. thesis, 1991, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign.

Xavier Serra, *A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition*. Department of Music Report No. STAN-M-58, Ph.D. dissertation, 1989, Center for Computer Music Research in Music and Acoustics, Stanford University.