Loudness Measurement according to EBU-R128

Fons Adriaensen
Casa della Musica, Parma

Linux Audio Conference 2011
Maynooth, Ireland

Where's that volume conrol ?

- Radio listeners and TV viewer hate it when they have to adjust the volume all the time. Why does this problem exist ?

  * The nature of contemporary broadcast content.
  * Decline of technical standards due to commercial pressure.
  * Automated play-out.
  * 'Out of context' production workflow.
  * The *Loudness Wars*.

- Broadcasters are aware of the problem.

What is missing ?

- Automated loudness measurement that

  * does not require human interpretation,
  * can be applied to stored data before it is used,
  * produces reliable results over e.g. a complete song, or a complete program.

- Technical standards.
- Consumer pressure and legislation.

Possible benefits

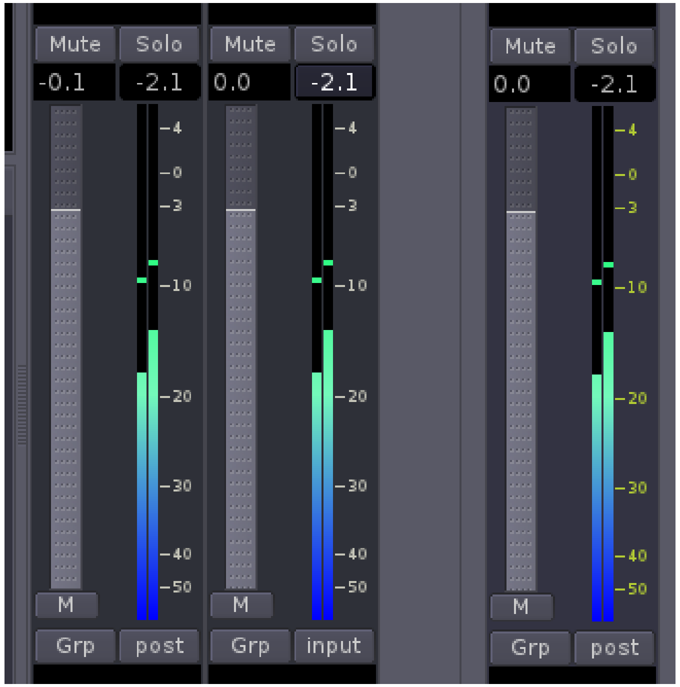
- A better listener and viewer experience.
- Maybe a solution for the Loudness Wars:

If broadcasters and consumer playback devices can measure loudness before playing a song and apply an automatic level correction, then there is nothing to be gained from pushing up recording levels at the expense of quality.

The state of the art

- Very few professional tools seem to exist (e.g. Dolby LM100).
- The film industry seems to have its act together.
- The music industry, and radio and TV broadcasters absolutely not.
- Current level measurement practices do not provide a solution.

Levels meters: Digital peak meter



- You all know these...
- Found in nearly all audio software.
- Measure peak sample value with slow fallback.
- Essential to check digital recording levels.
- No useful loudness indication at all.

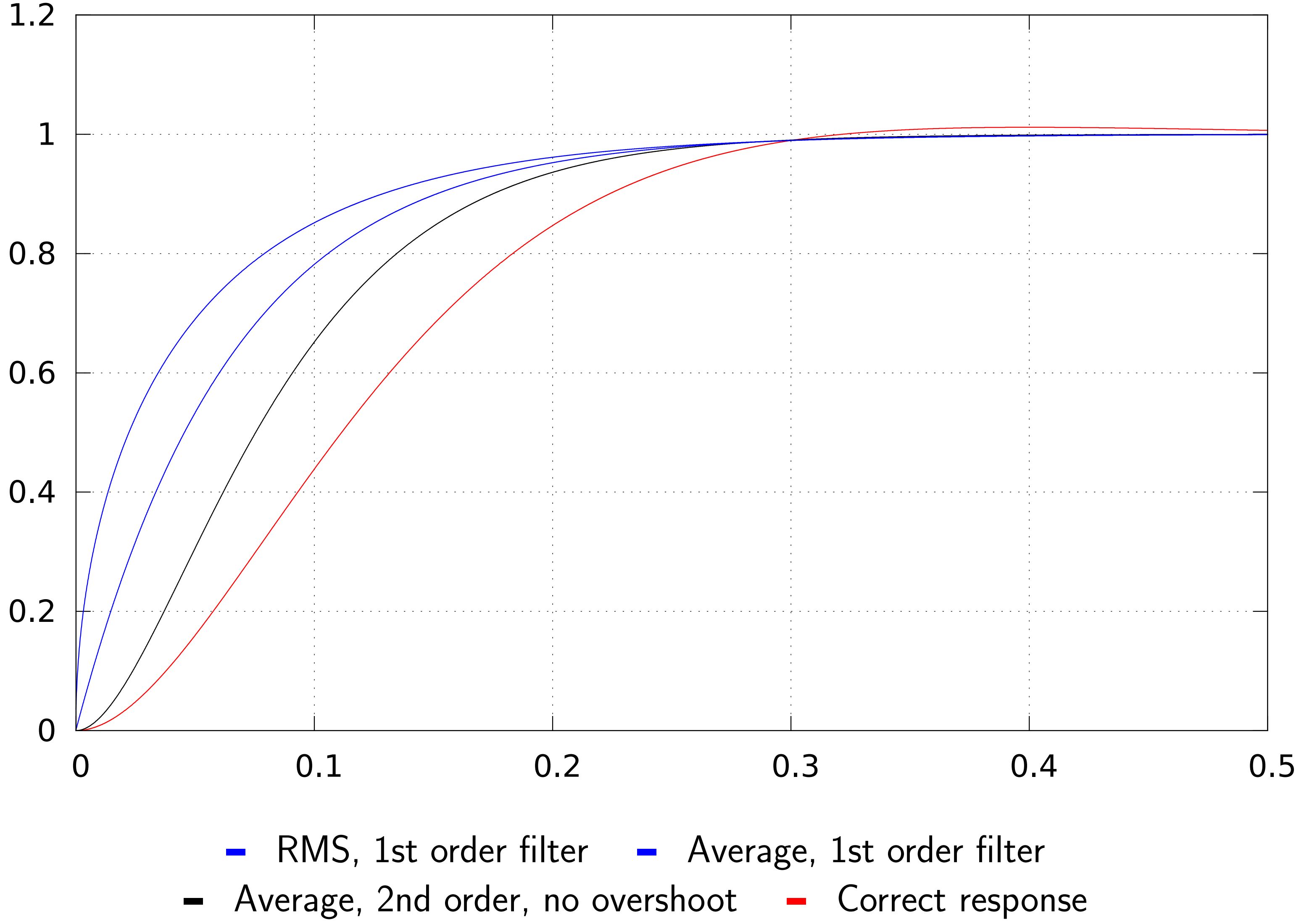
Levels meters: Pseudo peak meter

- BBC form shown, many others exist.
- Widely used in broadcasting (at least in Europe).
- Measure average level with fast rise time (10ms) and slow fall-back.
- Defined by international and corporate standards.
- Provide some indication of loudness but requires human interpretation.

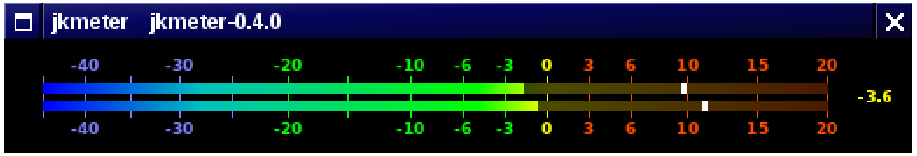
Levels meters: VU meter

- Popular in broadcasting and music studios in the U.S.A.
- Measures average of absolute value.
- Dynamic behaviour defined by the mechanics of the meter.
- Provide some indication of loudness but not very accurate.
- Standard form has limited dynamic range.
- Most software implementations get it wrong.

VU meter response



Level meters: K-meter



- Designed by mastering expert Bob Katz.
- Measures RMS and digital peak, displayed on the same scale.
- 'OdB' for RMS measurement offsett by 20 or 14 dB.
- Not widely used, but popularity is rising.
- No official standards.
- Provides *quite a good indication* of loudness.

Level meters: Summary

- With the exception of the K-meter, none of the currently used level meters provide a reliable loudness indication.

- VU and PMM require human interpretation and skilled users.

- They only provide *momentary* values, there is no standard for *integrated* measurement, nor for the determination of *loudness range*.

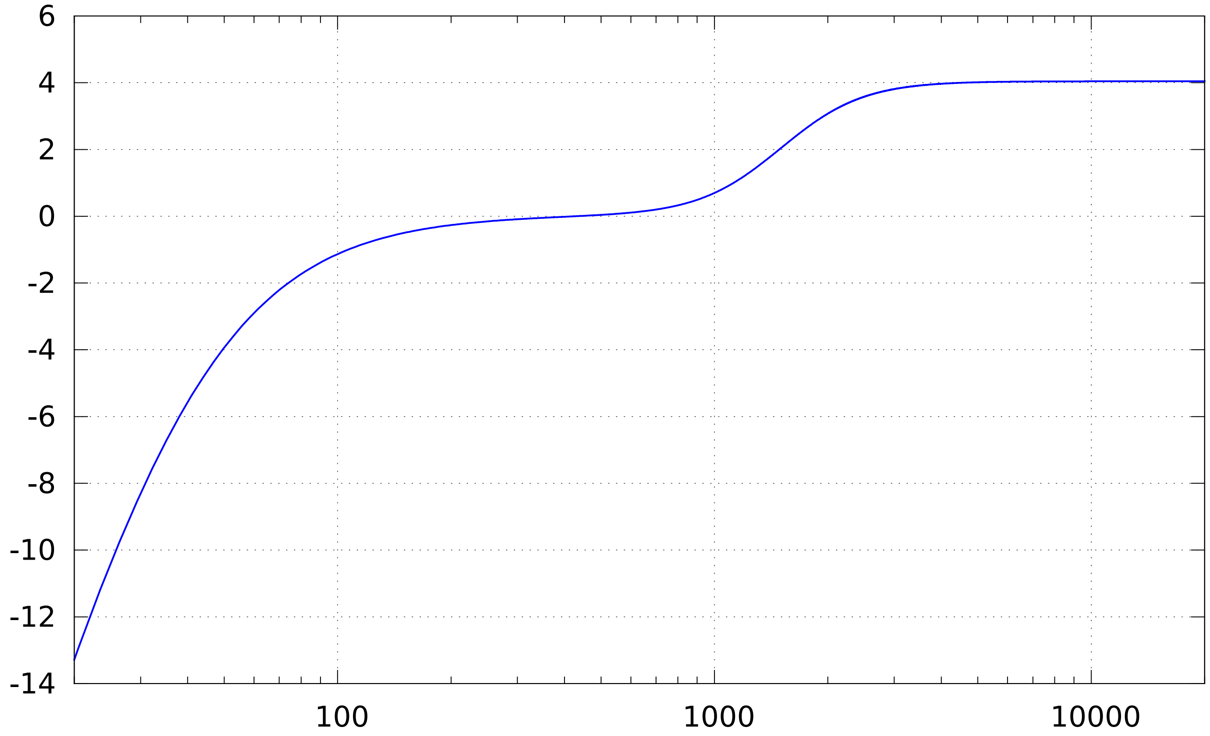
The EBU initiative

- Under the impulse of Florian Camerer (senior sound engineer at the ORF) the European Broadcasting Union has taken the initiative to define a loudness measurement standard.
- The *PLOUD* Working Group has produced Recommendation R-128, which is in turn based on standards defined by the ITU.
- R-128 defines methods to measure *integrated loudness* and *loudness range*, and some standards on how these values should be displayed.
- Implementation guidelines and audio test files are provided as well.
- Commercial equipment and software based on R-128 is starting to become available. Linux Audio should follow !

The ITU-R BS1770 recommendation

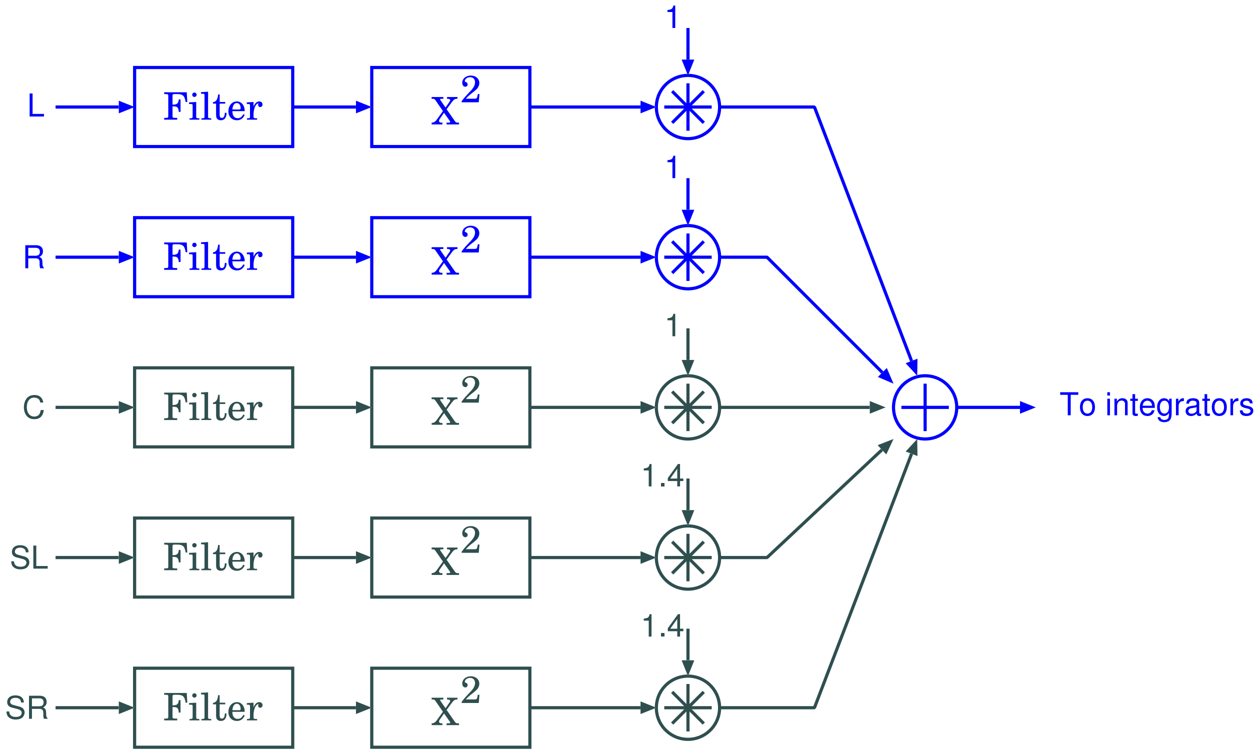
- R-BS1770 defines the basic loudness measurement algorithm. It is based on years of research by various institutions and members of the ITU.
- The algorithm has been validated by extensive listening tests which have shown very good correlation between measured and subjective results.
- Key features of the algorithm are:
  * Pre-detection filtering.
  * RMS measurement on individual channels.
  * Channel *powers* are summed.
  * Per-channel weights for 5.1 surround.
- The ITU recommendation does not specify a reference level.

The ITU filter



- Combination of second order highpass and second order shelf.
- Defined as a combination of two biquad sections.
- Gain needs to be normalised at 1kHz.

The ITU algorithm



The EBU R-128 recommendation

- R-128 builds on the ITU recommendation and defines the the algorithms to compute four ouputs:
  - **M** : momentary loudness
  - **S** : short term loudness
  - **I** : integrated loudness
  - **LRA** : loudness range
- It also specifies the reference level as **-23 dB** w.r.t. a sine wave at maximum digital level.
- Measured values should be displayed either as *absolute*, using the unit **LUFS**, or relative to the reference level and using the unit **LU**. For both, this unit has the same meaning as the dB.
- R-128 also defines the ranges and update rates for the display, some required controls and annotation, but not e.g. the exact layout or colors.

The **M** and **S** measurements

- The **M** ouput is computed by integrating the sum of powers over a sliding rectangular window of 400 ms.
- The **S** ouput is computed by integrating the sum of powers over a sliding rectangular window of 3 seconds.
- An instrument or software application conforming to R-128 should allow the user to display either **M** or **S**. Update rate must be at least 10 times per second.
- The maximum values for either must be displayed as well.
- For a graphical display two ranges should be provided: one from -18 to +9 LU, the second from -36 to +18 LU.
- The user should be able to select either the relative (LU) scale, or the absolute (LUFS) one.

The integrated loudness algorithm

- The **I** algorithm provides a loudness value averaged over an arbitrary long time interval.
- It can be applied to either an audio file, or in interactive mode, controlled by *start*, *stop* and *reset* commands. The same controls also enable and reset the display of the maximum values of **M** or **S**.
- The measurement is based on the 400 ms windows used for the **M** display. The integration periods must overlap by at least 200 ms.
- Given this input, the integrated loudness is computed in four steps:
  - * All inputs below -70 dB (re. full scale) are discarded.
  - * The average power of the remaining values is computed.
  - * All inputs lower than 8 dB below this average are discarded.
  - * The output is the average power of the remaining values.

The loudness range algorithm - 1

- The **LRA** algorithm provides an indication of the *loudness range* over an arbitrary long time interval. It can be used e.g. to determine if some compression is required.
- It can be applied to either an audio file, or in interactive mode, using the same controls as for the integrated measurement.
- The algorithm is designed to ignore periods of almost silence, and very loud but short sounds (e.g. a gunshot in a movie).
- R-128 does not require endpoints of the range to be displayed, only their difference.
- The measurement is based on the 3 second windows used for the **S** display. The integration periods must overlap by at least 2 seconds.

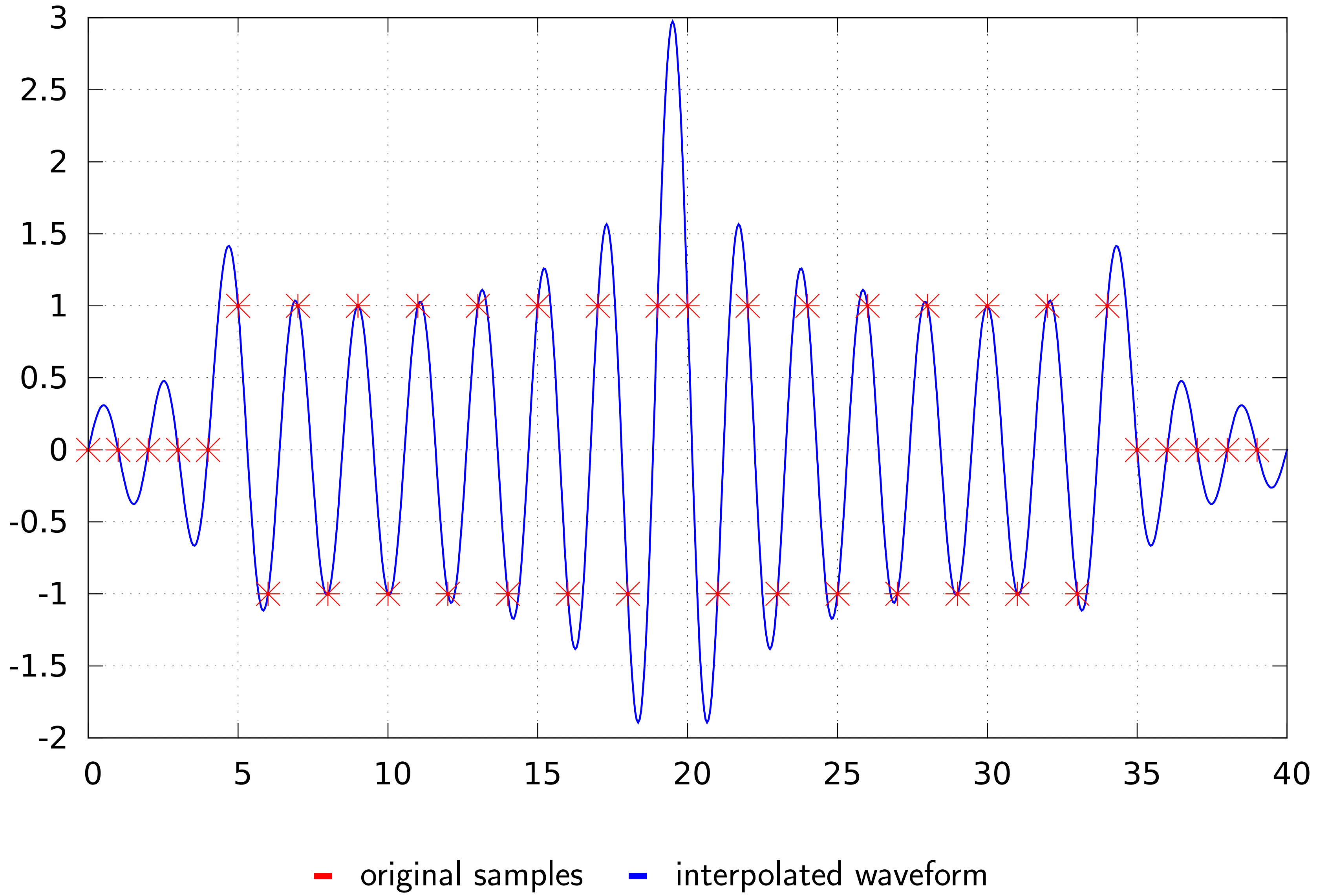
The loudness range algorithm - 2

- Given the values used for the **S** measurement, the loudness range is calculated in four steps:
  * All values below -70 dB (re. full scale) are discarded.
  * The average power of the remaining inputs is computed.
  * All inputs lower than 20 dB below this average are discarded.
  * The loudness range is then computed as the difference between the level that is exceeded by 90% of the remaining values, and the one exceeded by 5% of them.
- The use of *percentiles* to determine the final output provides for most 'robust' estimation.

Peak indicators

- EBU-R128 also requires an indication of peak value going above limits.
- Inter-sample values easily exceed sample values.
- The ITU documents suggest to upsample by at least a factor of four to find the real peak values.

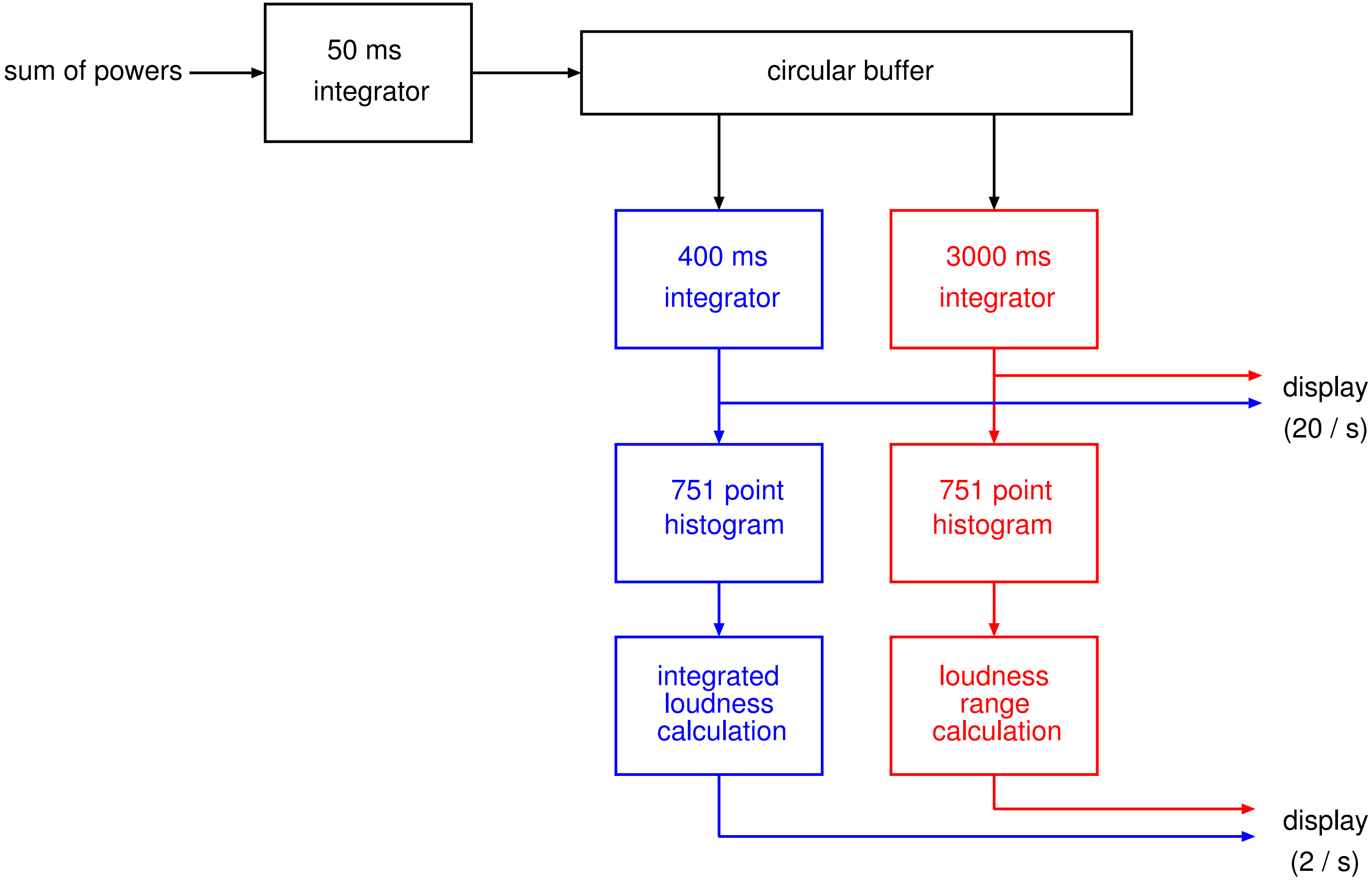
Inter-sample peaks



Implementation of the EBU-R128 algorithms - 1

- A naive implementation (and the Mathlab code provided by the EBU) would require unbounded storage, and the complexity of the calculation would increase as the time interval becomes longer.
- Both problems can be avoided by the use of *histogram* data structures:
  - * Fixed data size for any length of the history.
  - * Efficient fixed-time calculation of averages and percentiles.
- The implementation presented uses two histograms with 751 'bins' each, covering the range -70 to +5 dB with a step of 0.1 dB. A small step size removes the need for interpolation in the histogram data.
- The input data rate for the **I** and **LRA** algorithms is twice the minumum required by R-128.

Implementation of the EBU-R128 algorithms - 2



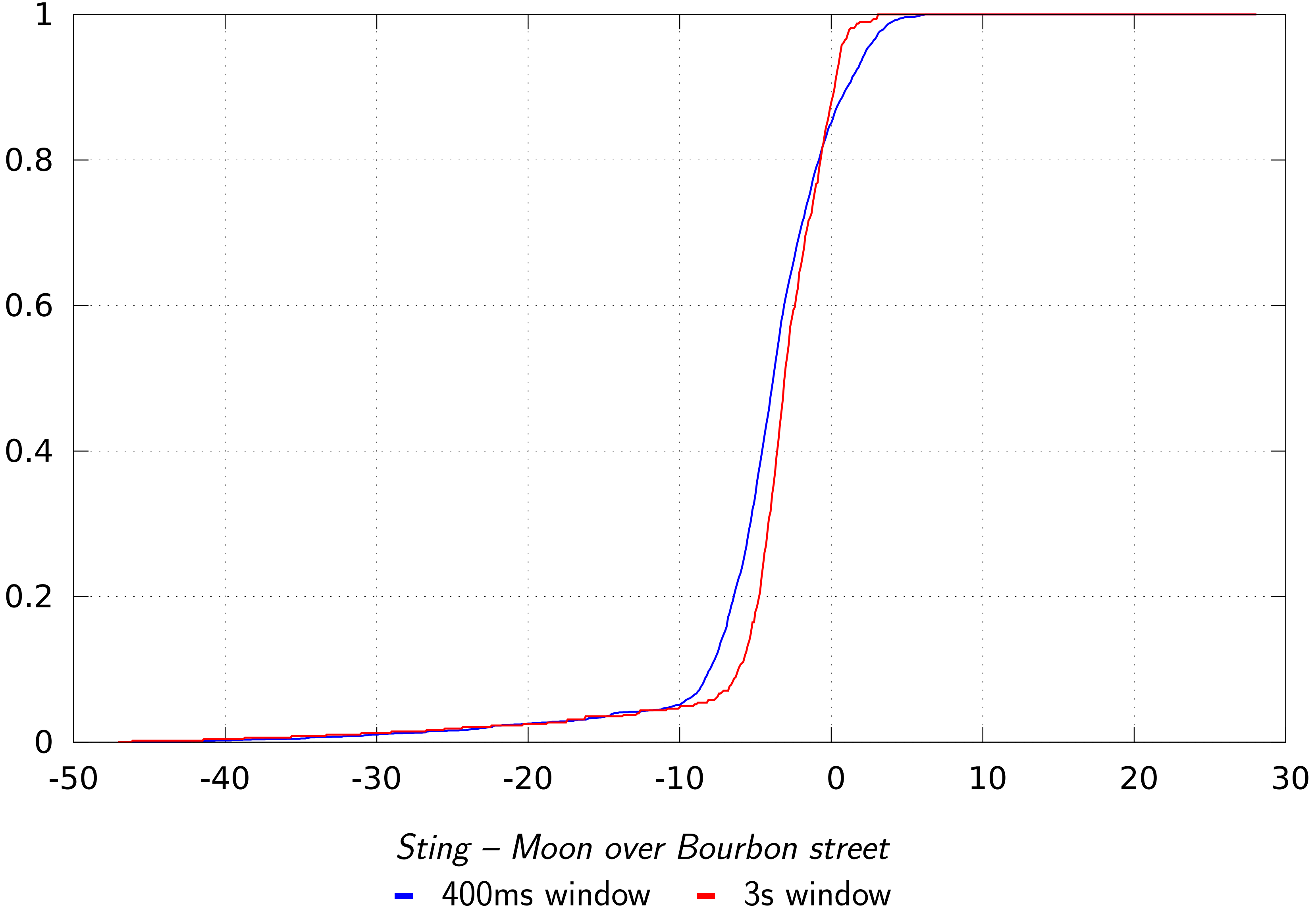
The Linux programs

- Two applications are provided in the first release:
- ebumeter
  * Real-time measurment with GUI.
  * First release does not have peak indicators.
  * Future version may include graphical display of loudness history.
- ebur128
  * Command line app to measure audio files.
  * Provides some extra information (internal values).
  * Optionally produces a cumulative probability data file that can be displayed using Gnuplot.

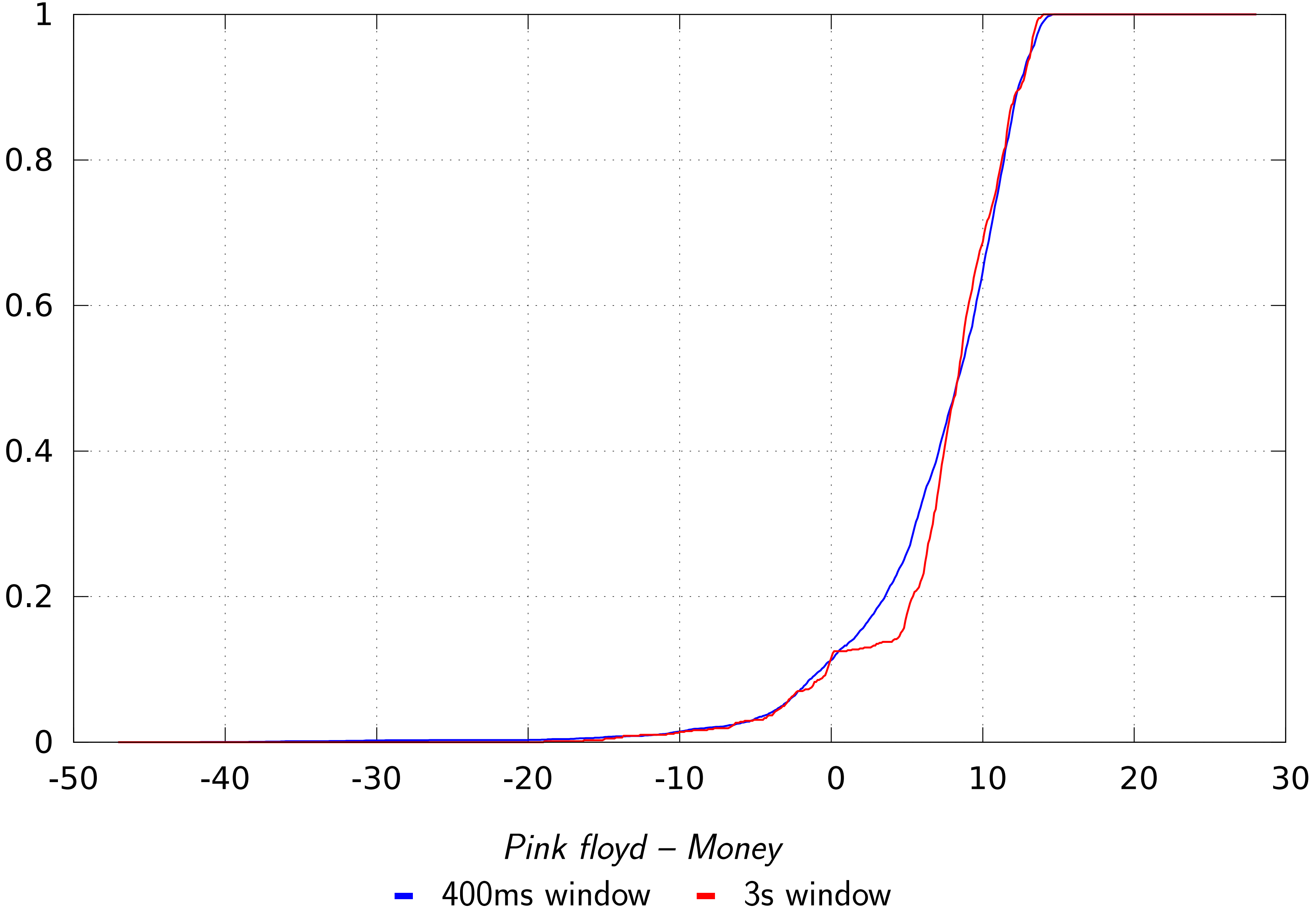
Some results

Title	I	LRA	pk	pk-I
Sting - Moon over Bourbon street	-2.1	6.3	-8.2	16.9
Weather Report - A remark you made	+6.1	11.1	-1.1	15.8
D. Fagen - Maxine	+2.0	7.1	-3.6	17.4
Pink Floyd - Money	+9.8	13.3	-0.4	12.8
Sheryl Crow - All I wanna do	+7.4	2.7	0.0	15.6
F. Mendelssohn - Die Hebriden	+4.0	19.9	-0.3	18.7
D. Shostakovich String Quartet 3	+6.0	14.0	-1.1	15.9
B. Britten - Sea Interludes	+4.0	18.7	-0.1	18.9
A. Bruckner - Symphony 9 - 1	+6.2	24.4	0.0	16.8
A. Bruckner - Symphony 9 - 2	+6.9	23.8	0.0	16.1

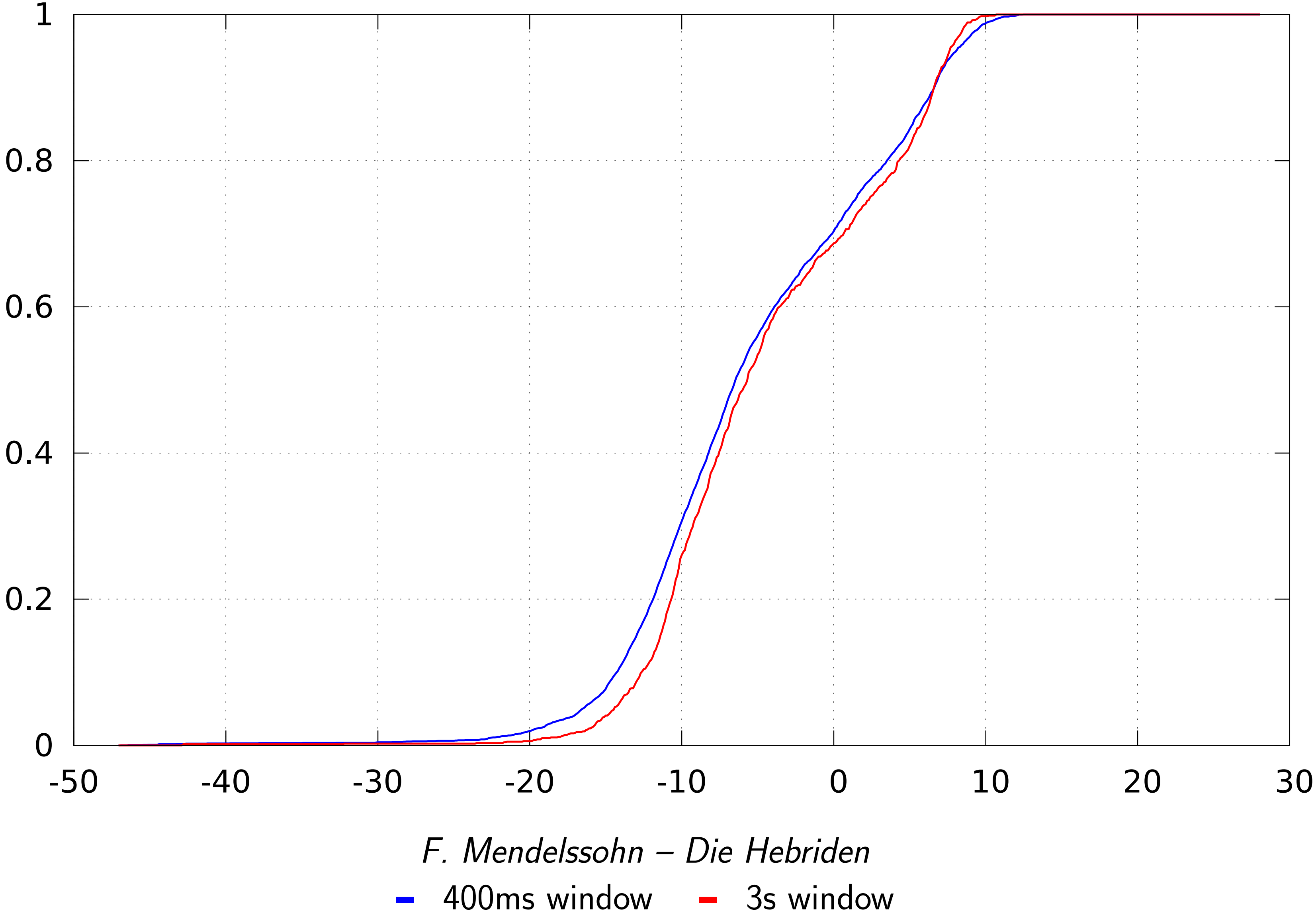
Probability distribution - 1



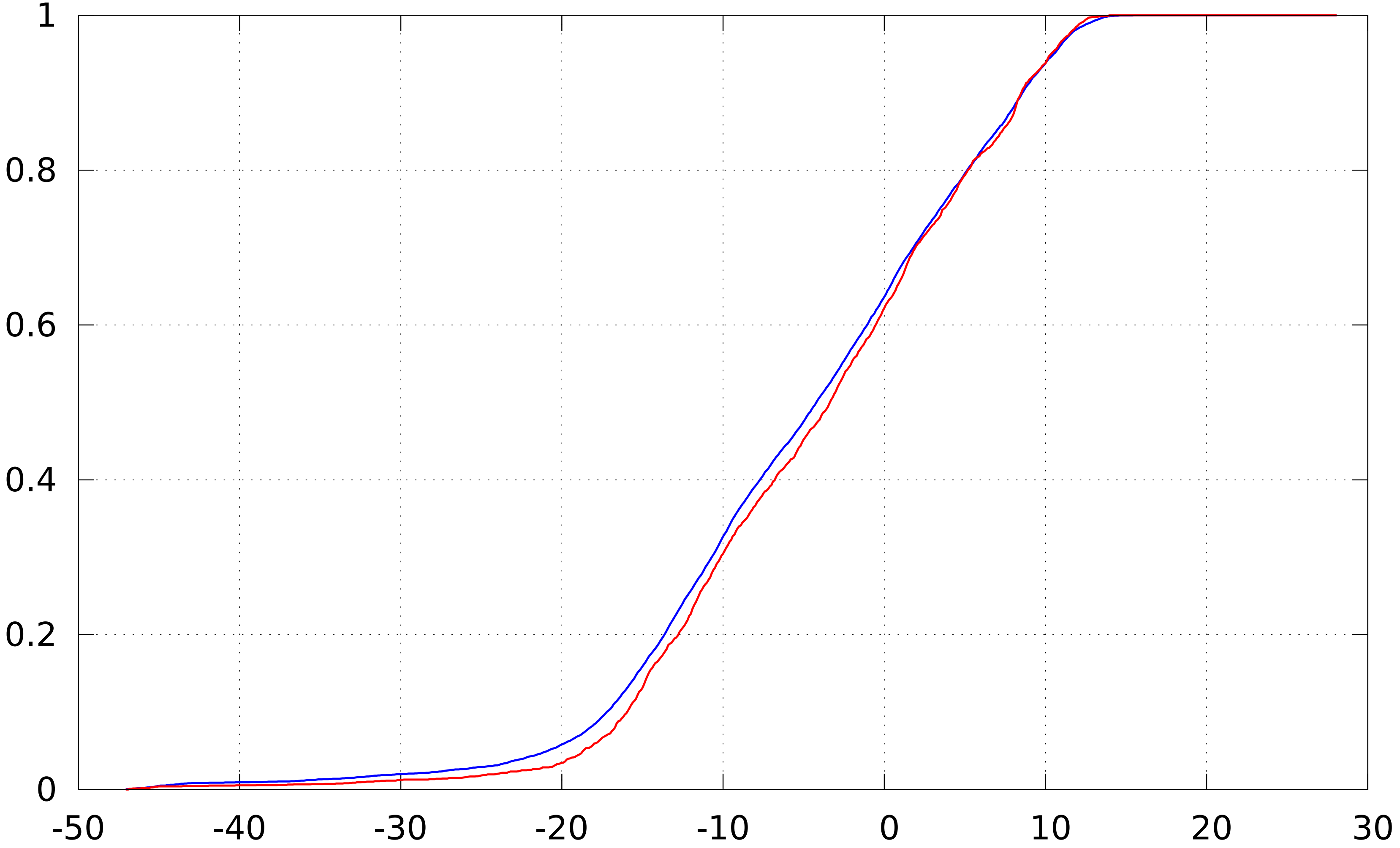
Probability distribution - 2



Probability distribution - 3



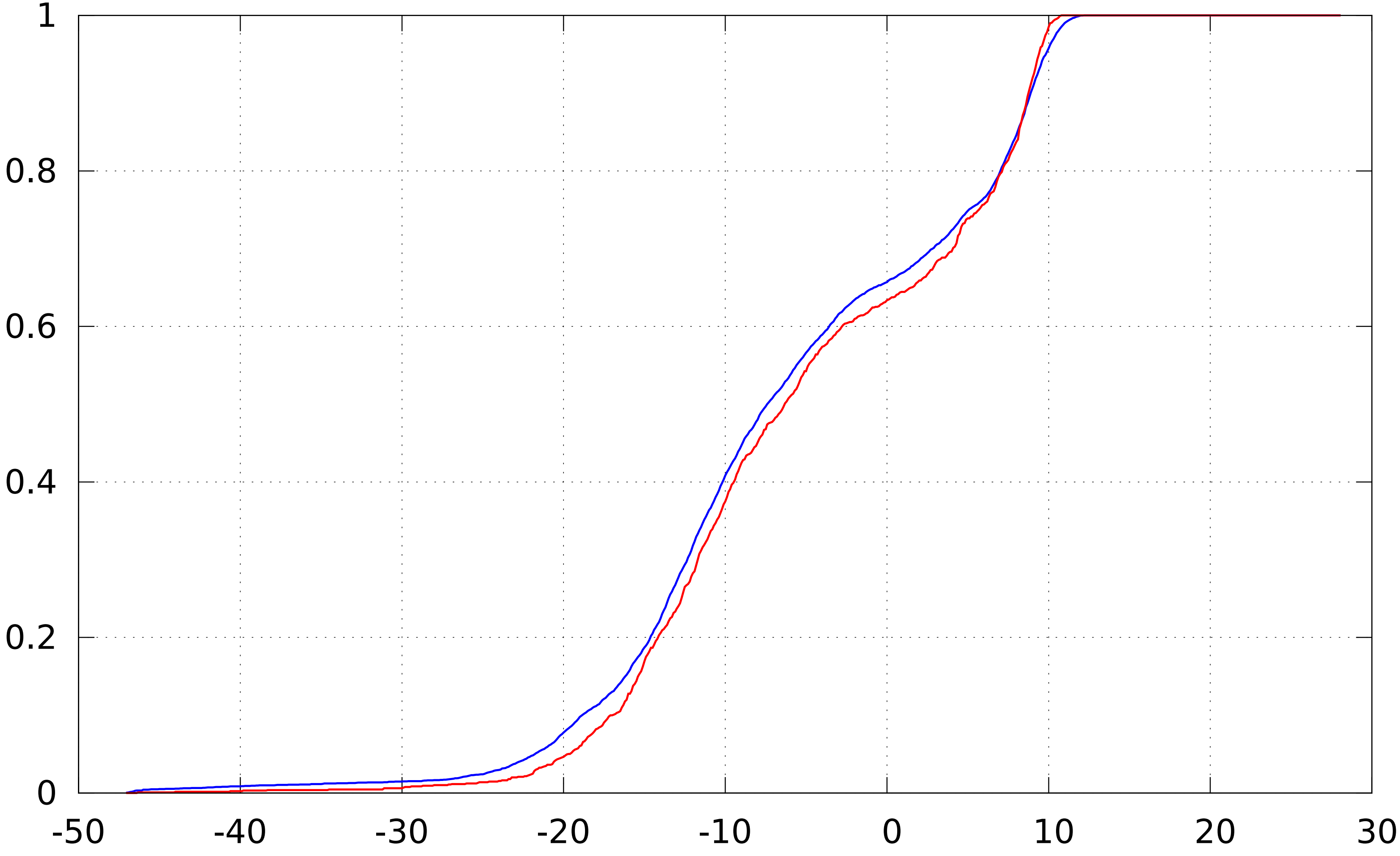
Probability distribution - 4



A. Bruckner – Symphony No. 9, Feierlich, misterioso

400ms window   3s window

Probability distribution - 5



A. Bruckner – Symphony No. 9, Bewegt, lebhaft

400ms window   3s window

Questions and answers

The end