Application of Linux Audio in Hearing Aid Research

Giso Grimm\textsuperscript{1} \quad Tobias Herzke\textsuperscript{2} \quad Volker Hohmann\textsuperscript{2}

\textsuperscript{1}Universität Oldenburg, Oldenburg, Germany

\textsuperscript{2}HörTech gGmbH, Oldenburg, Germany

Linux Audio Conference, April 17th, 2009.
Introduction and Overview

Hearing Aids
Algorithms for Digital Hearing Aids
Evaluation of Hearing Aids

A Linux-based Hearing Aid

RT performance
CPU and battery performance
Delay Constraints

Conclusions
Hearing Aids

Analogue hearing aids

- Limited capabilities:
- Frequency shaping and amplification
- Dynamic compression
- Static notch filters for feedback cancellation
Hearing Aids

Analogue hearing aids

- Limited capabilities:
- Frequency shaping and amplification
- Dynamic compression
- Static notch filters for feedback cancellation

Digital hearing aids

- First hearing aid with Digital Signal Processing in 1996
- New algorithms with no analogue counterpart
- Wireless binaural link between ears
  - Parameter exchange
  - Coming soon: Low-delay low-power audio transmission
Limitations of hearing aids

Device
- Small battery capacity
- Low processing power

Acoustics
- Acoustic feedback
- Limited audio bandwidth (by tube between receiver and ear)

End user
- High expenses
- Unfulfilled expectations
Algorithms scalable to individual hearing loss

- Dynamic compression
- Frequency shaping
- Frequency compression

Signal enhancement

- Directional microphones
- Noise reduction

Artifact reduction

- Feedback cancellation
- Future: Non-linearity compensation, ...
Why dynamic compression?

ISO normal HTL

Normal hearing threshold
Why dynamic compression?

- Hearing impaired threshold
- Amplification and frequency shaping
Why dynamic compression?

- Normal uncomfortable level
Why dynamic compression?

- Hearing impaired uncomfortable level
- Dynamic compression
Directional microphones: SNR improvement

- Delay-and-sum: up to 3 dB SNR-improvement for on-axis sounds in diffuse noise
- Many more beamformers and direction-of-arrival estimators
Noise reduction and de-reverberation

- Single channel noise reduction
- Binaural noise reduction, de-reverberation
- Binaural coherence estimation: Interaural phase difference statistics
Artifact reduction: Feedback cancellation

- Adaptive feedback cancellation
- Estimation of feedback signal
- Subtraction from input

![Diagram of Hearing Aid System](image)
Artifact reduction: Feedback cancellation

- Adaptive feedback cancellation
- Estimation of feedback signal
- Subtraction from input
- Other solutions: Frequency shifting, phase modulation, binaural coherence
Evaluation of Hearing Aids

Expected benefit of hearing aids

- Improvement of speech intelligibility
- Reduction of listening effort in adverse listening conditions
- Increase of listening comfort
Evaluation of Hearing Aids

Expected benefit of hearing aids

- Improvement of speech intelligibility
- Reduction of listening effort in adverse listening conditions
- Increase of listening comfort

Evaluation is required!
Evaluation methods

'Objective’ methods

- SNR improvement (e.g., shadow filtering)
- Speech intelligibility index ($f$-weighted SNR)
- Quality prediction (usually similarity measures)

Subjective methods

- Speech recognition threshold in quiet and noise
- Quality rating, paired comparison ...
Test conditions

- ‘static’ situations:
  no spatial influence, pre-processed or real-time processing

- dynamic conditions:
  head movements or moving sources, real-time processing

- realistic devices, acoustic feedback:
  low-delay real-time processing

- Real-life conditions:
  wide dynamic range, portable low-delay real-time processing
Real-time systems

1995: PC with DSP board, assembler programming
Real-time systems

Real-time systems

2008: Portable PC, C++ (or Matlab) programming
A Linux Hearing Aid
A Linux Hearing Aid

for research, development and field testing
The 'Master Hearing Aid' software

- Platform for hearing aid algorithm development and evaluation
- Many audio backends: JACK, ALSA (Linux), ASIO (Windows), file, network, Matlab
- Extremely modular structure
- Hearing-aid (and hearing) related processing blocks
- Commercial product (closed source) by HoerTech gGmbH, Oldenburg
- Used in research projects and hearing aid industry
Licensing in competitive research

- Industrial/competitive context often requires closed-source development
- Industry decision makers fear the open source licensing of Linux
- It is possible to develop closed software on the Linux platform without infringing any licenses!
- Careful consideration of what components to use
- Knowledge of the relevant licenses, and compliance with their terms.
- Use of open source software in industrial context beneficial for both sides
Comparison of RT kernel versus non-RT kernel

Kernel version 2.6.X (2.6.23?), from Grimm et al. (2006)
RT performance and low-delay

Algorithm implementation in C/C++
RT performance and low-delay

Algorithm implementation in C/C++

Algorithm implementation in Matlab
CPU and battery performance

- Two portable systems
- Asus Eee PC 701:
  Intel Celeron @ 630 MHz
- Acer Aspire one:
  Intel Atom @ 1.6 GHz
- Reference Desktop system (Intel P4 @ 3GHz)
Comparison of CPU performance
Comparison of batteries

The graph compares the battery runtime for different algorithms (algo1, algo2, algo3, algo4) for two devices: Asus Eee PC 701 and Acer Aspire One. The runtime is measured in hours (2h00', 2h15', 2h30', 2h45', 3h00', 3h15', 3h30').

- Asus Eee PC 701:
  - algo1: 3h30'
  - algo2: 3h15'
  - algo3: 3h00'
  - algo4: 2h30'

- Acer Aspire One:
  - algo1: 3h00'
  - algo2: 2h30'
  - algo3: 2h15'
  - algo4: 2h00'
Delay Constraints

Three sources of delay in RT processing

- **Block processing delay:**
  Fragment size plus processing delay
  (i.e., two blocks if input and output is block aligned)

- **Hardware delay:**
  Anti-aliasing filter (AD and DA, typical 1.5 ms in total)
  Data transmission, driver layer (‘mystic’ delays)

- **Algorithmic delay:**
  Group delay of filters, overlap-add delay, ...
Why do we aim for lowest delay? ⇒ Feedback howling

- Feedback criterion: Roundtrip-gain is above 0 dB, and phase is multiple of $2\pi$
Why do we aim for lowest delay? ⇒ Feedback howling

- Feedback criterion: Roundtrip-gain is above 0 dB, and phase is multiple of \(2\pi\)
## Achievable delays

<table>
<thead>
<tr>
<th>Device</th>
<th>$f_s$/kHz</th>
<th>fragment size</th>
<th>$\tau_{sc}$/ms</th>
<th>$\tau_t$/ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Echo Layla 3G</td>
<td>32</td>
<td>32</td>
<td>2.81</td>
<td>4.81</td>
</tr>
<tr>
<td></td>
<td>44.1</td>
<td>64</td>
<td>2.04</td>
<td>4.94</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>32</td>
<td>0.81</td>
<td>1.77</td>
</tr>
<tr>
<td>RME HDSP9652 + Behringer SRC2496</td>
<td>32</td>
<td>64</td>
<td>3.34</td>
<td>7.34</td>
</tr>
<tr>
<td></td>
<td>44.1</td>
<td>64</td>
<td>2.68</td>
<td>5.58</td>
</tr>
<tr>
<td></td>
<td>96</td>
<td>128</td>
<td>1.73</td>
<td>4.40</td>
</tr>
<tr>
<td>RME HDSP9652 + Behringer ADA8000</td>
<td>32</td>
<td>64</td>
<td>2.13</td>
<td>6.12</td>
</tr>
<tr>
<td></td>
<td>44.1</td>
<td>64</td>
<td>1.61</td>
<td>4.51</td>
</tr>
<tr>
<td>RME HDSP9632 + ADI8QS</td>
<td>44.1</td>
<td>64</td>
<td>1.52</td>
<td>4.42</td>
</tr>
<tr>
<td></td>
<td>96</td>
<td>128</td>
<td>1.03</td>
<td>3.7</td>
</tr>
<tr>
<td>OFFIS</td>
<td>16</td>
<td>16</td>
<td>6.81</td>
<td>9.81</td>
</tr>
<tr>
<td>USB SC-4/2</td>
<td>44.1</td>
<td>64</td>
<td>4.08</td>
<td>8.44</td>
</tr>
<tr>
<td></td>
<td>96</td>
<td>128</td>
<td>1.97</td>
<td>5.97</td>
</tr>
</tbody>
</table>
### RT performance

<table>
<thead>
<tr>
<th>Device</th>
<th>$f_s$/kHz</th>
<th>fragment size</th>
<th>$\tau_{sc}$/ms</th>
<th>$\tau_t$/ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Echo Layla 3G</td>
<td>32</td>
<td>32</td>
<td>2.81</td>
<td>4.81</td>
</tr>
<tr>
<td></td>
<td>44.1</td>
<td>64</td>
<td>2.04</td>
<td>4.94</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>32</td>
<td>0.81</td>
<td>1.77</td>
</tr>
<tr>
<td>RME HDSP9652 +</td>
<td>32</td>
<td>64</td>
<td>3.34</td>
<td>7.34</td>
</tr>
<tr>
<td>Behringer SRC2496</td>
<td>44.1</td>
<td>64</td>
<td>2.68</td>
<td>5.58</td>
</tr>
<tr>
<td></td>
<td>96</td>
<td>128</td>
<td>1.73</td>
<td>4.40</td>
</tr>
<tr>
<td>RME HDSP9652 +</td>
<td>32</td>
<td>64</td>
<td>2.13</td>
<td>6.12</td>
</tr>
<tr>
<td>Behringer ADA8000</td>
<td>44.1</td>
<td>64</td>
<td>1.61</td>
<td>4.51</td>
</tr>
<tr>
<td>RME HDSP9632 +</td>
<td>44.1</td>
<td>64</td>
<td>1.52</td>
<td>4.42</td>
</tr>
<tr>
<td>ADI8QS</td>
<td>96</td>
<td>128</td>
<td>1.03</td>
<td>3.7</td>
</tr>
<tr>
<td>OFFIS</td>
<td>16</td>
<td>16</td>
<td>6.81</td>
<td>9.81</td>
</tr>
<tr>
<td>USB SC-4/2</td>
<td>44.1</td>
<td>64</td>
<td>4.08</td>
<td>8.44</td>
</tr>
<tr>
<td></td>
<td>96</td>
<td>128</td>
<td>1.97</td>
<td>5.97</td>
</tr>
</tbody>
</table>
## Device Performance Comparison

<table>
<thead>
<tr>
<th>Device</th>
<th>$f_s$/kHz</th>
<th>fragment size</th>
<th>$\tau_{sc}$/ms</th>
<th>$\tau_t$/ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Echo Layla 3G</td>
<td>32</td>
<td>32</td>
<td>2.81</td>
<td>4.81</td>
</tr>
<tr>
<td></td>
<td>44.1</td>
<td>64</td>
<td>2.04</td>
<td>4.94</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>32</td>
<td>0.81</td>
<td>1.77</td>
</tr>
<tr>
<td>RME HDSP9652 + Behringer SRC2496</td>
<td>32</td>
<td>64</td>
<td>3.34</td>
<td>7.34</td>
</tr>
<tr>
<td></td>
<td>44.1</td>
<td>64</td>
<td>2.68</td>
<td>5.58</td>
</tr>
<tr>
<td></td>
<td>96</td>
<td>128</td>
<td>1.73</td>
<td>4.40</td>
</tr>
<tr>
<td>RME HDSP9652 + Behringer ADA8000</td>
<td>32</td>
<td>64</td>
<td>2.13</td>
<td>6.12</td>
</tr>
<tr>
<td></td>
<td>48</td>
<td>64</td>
<td>1.46</td>
<td>4.12</td>
</tr>
<tr>
<td>RME HDSP9632 + ADI8QS</td>
<td>44.1</td>
<td>64</td>
<td>1.52</td>
<td>4.42</td>
</tr>
<tr>
<td></td>
<td>96</td>
<td>128</td>
<td>1.03</td>
<td>3.7</td>
</tr>
<tr>
<td>OFFIS</td>
<td>16</td>
<td>16</td>
<td>6.81</td>
<td>9.81</td>
</tr>
<tr>
<td>USB SC-4/2</td>
<td>44.1</td>
<td>64</td>
<td>4.08</td>
<td>8.44</td>
</tr>
<tr>
<td></td>
<td>96</td>
<td>128</td>
<td>1.97</td>
<td>5.97</td>
</tr>
</tbody>
</table>
## RT performance

<table>
<thead>
<tr>
<th>Device</th>
<th>$f_s$/kHz</th>
<th>fragment size</th>
<th>$\tau_{sc}$/ms</th>
<th>$\tau_t$/ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Echo Layla 3G</td>
<td>32</td>
<td>32</td>
<td>2.81</td>
<td>4.81</td>
</tr>
<tr>
<td></td>
<td>44.1</td>
<td>64</td>
<td>2.04</td>
<td>4.94</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>32</td>
<td>0.81</td>
<td>1.77</td>
</tr>
<tr>
<td>RME HDSP9652 + Behringer SRC2496</td>
<td>32</td>
<td>64</td>
<td>3.34</td>
<td>7.34</td>
</tr>
<tr>
<td></td>
<td>44.1</td>
<td>64</td>
<td>2.68</td>
<td>5.58</td>
</tr>
<tr>
<td></td>
<td>96</td>
<td>128</td>
<td>1.73</td>
<td>4.40</td>
</tr>
<tr>
<td>RME HDSP9652 + Behringer ADA8000</td>
<td>32</td>
<td>64</td>
<td>2.13</td>
<td>6.12</td>
</tr>
<tr>
<td></td>
<td>44.1</td>
<td>64</td>
<td>1.61</td>
<td>4.51</td>
</tr>
<tr>
<td>RME HDSP9632 + ADI8QS</td>
<td>44.1</td>
<td>64</td>
<td>1.52</td>
<td>4.42</td>
</tr>
<tr>
<td></td>
<td>96</td>
<td>128</td>
<td>1.03</td>
<td>3.7</td>
</tr>
<tr>
<td>OFFIS</td>
<td>16</td>
<td>16</td>
<td>6.81</td>
<td>9.81</td>
</tr>
<tr>
<td>USB SC-4/2</td>
<td>44.1</td>
<td>64</td>
<td>4.08</td>
<td>8.44</td>
</tr>
<tr>
<td></td>
<td>96</td>
<td>128</td>
<td>1.97</td>
<td>5.97</td>
</tr>
</tbody>
</table>
## Device Specifications

<table>
<thead>
<tr>
<th>Device</th>
<th>$f_s$/kHz</th>
<th>fragment size</th>
<th>$\tau_{sc}$/ms</th>
<th>$\tau_t$/ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Echo Layla 3G</td>
<td>32</td>
<td>32</td>
<td>2.81</td>
<td>4.81</td>
</tr>
<tr>
<td></td>
<td>44.1</td>
<td>64</td>
<td>2.04</td>
<td>4.94</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>32</td>
<td>0.81</td>
<td>1.77</td>
</tr>
<tr>
<td>RME HDSP9652 + Behringer SRC2496</td>
<td>32</td>
<td>64</td>
<td>3.34</td>
<td>7.34</td>
</tr>
<tr>
<td></td>
<td>44.1</td>
<td>64</td>
<td>2.68</td>
<td>5.58</td>
</tr>
<tr>
<td></td>
<td>96</td>
<td>128</td>
<td>1.73</td>
<td>4.40</td>
</tr>
<tr>
<td>RME HDSP9652 + Behringer ADA8000</td>
<td>32</td>
<td>64</td>
<td>2.13</td>
<td>6.12</td>
</tr>
<tr>
<td></td>
<td>44.1</td>
<td>64</td>
<td>1.61</td>
<td>4.51</td>
</tr>
<tr>
<td>RME HDSP9632 + ADI8QS</td>
<td>44.1</td>
<td>64</td>
<td>1.52</td>
<td>4.42</td>
</tr>
<tr>
<td></td>
<td>96</td>
<td>128</td>
<td>1.03</td>
<td>3.7</td>
</tr>
<tr>
<td>OFFIS</td>
<td>16</td>
<td>16</td>
<td>6.81</td>
<td>9.81</td>
</tr>
<tr>
<td>USB SC-4/2</td>
<td>44.1</td>
<td>64</td>
<td>4.08</td>
<td>8.44</td>
</tr>
<tr>
<td></td>
<td>96</td>
<td>128</td>
<td>1.97</td>
<td>5.97</td>
</tr>
</tbody>
</table>
Demonstration now!
Conclusions

- Hearing aid research is a wide field and requires careful evaluation.
- Linux Audio provides a valuable environment for hearing aid research.
- Low delay processing is possible with Linux/ALSA/Jack (but mystic delays remain in the chain of soft- and hardware).
Thank you for your attention!