A skim through HARK, a modular OSS system for robot audition

Alexandre Trilla

October 2011
1 Introduction to HARK

2 HARK approaches to noise reduction

3 HARK modular framework

4 Application of HARK
1. Introduction to HARK

2. HARK approaches to noise reduction

3. HARK modular framework

4. Application of HARK
Introduction to HARK

**Robot audition**: robot capability of listening to several pieces of speech at once by itself

- **Critical issue**: Real-Time (RT) processing in a **noisy environment**
  - Near-end speech (headset microphone) → **OK**
    - High Signal-to-Noise Ratio (SNR)
  - Far-end speech (distant speaker) → **KO**
    - Low SNR (attenuated speech signal + additive noise)
    - Typical Automatic Speech Recognition (ASR) fails mainly due to the **single channel limitation**
1 Introduction to HARK

2 HARK approaches to noise reduction

3 HARK modular framework

4 Application of HARK
HARK approaches to noise reduction

ASR preprocessing through microphone-array-based techniques
→ Signal space diversity → SNR increase
  ■ Directional noise → Sound source localisation and separation
    ▪ if (direction of arrival between two sources > 20° and
      #microphones > #sources), then every source can be
      separated (in theory)
  ■ Diffuse noise (e.g., babble noise) → Speech Enhancement
    ▪ Does not include direction explicitly
  ■ Reverb noise (acoustic enclosure)
    ▪ Early (intra-frame in ASR, ~ 25ms) → Acoustic model
      improvement
    ▪ Late (inter-frame in ASR, ~ 200ms) → No fixed model can be
      assumed → Adaptive filtering
  ■ Ego noise (internal) → Template-based method using joint
    status info (to be implemented)
ASR of separated speech → Time-frequency map of reliability to
be robust against spectral distortion due to separation
Outline

1. Introduction to HARK
2. HARK approaches to noise reduction
3. HARK modular framework
4. Application of HARK
HARK modular framework

- **Dataflow programming** with FlowDesigner as middleware
  - Network of modules connected dynamically at runtime
  - Well balanced trade-off between independence and processing speed

- **Functional modules**
  [Nakadai et al., 2008, Nakadai et al., 2011]
  - **MUSIC:** Adaptive beamformer for sound source localisation
    - Compromise between robustness for environment change and peak performance
  - **GHDSS-AS:** Hybrid beamformer and Blind Source Separation for sound source separation
    - Geometric constraints obtained from the locations of the microphones and sound sources

- **HRLE:** Histogram-based method for Speech Enhancement
- **MFT-ASR:** Feature masks to cope with distortions for ASR
Multichannel audio device: usually 8ch., although HARK does not specify any number

Online & RT processing (30ms frame + 10ms overlap)
- Module processing time < 10ms → 3750 computer instructions (single core μ-P, 1.5GHz, 4 CPI, optimised C/C++ compiler) for processing 480 samples (30ms frame at 16KHz) seems feasible

Function-call based integration with ManyEars
- Steered beamforming, particle-filtering-based tracking, etc.
Outline

1. Introduction to HARK
2. HARK approaches to noise reduction
3. HARK modular framework
4. Application of HARK
Application of HARK

- **Dialogue system** based on a deterministic Finite-State Automaton, e.g., rock-paper-scissors game referee [Nakadai et al., 2011]
- **Dialogue system** based on simple heuristics, e.g., meal order taking [Nakadai et al., 2011]


A skim through HARK, a modular OSS system for robot audition
A skim through HARK, a modular OSS system for robot audition

Alexandre Trilla

October 2011