JOINT EGO-NOISE SUPPRESSION AND KEYWORD SPOTTING ON SWEEPING ROBOTS

Yueyue Na, Ziteng Wang, Liang Wang, Qiang Fu

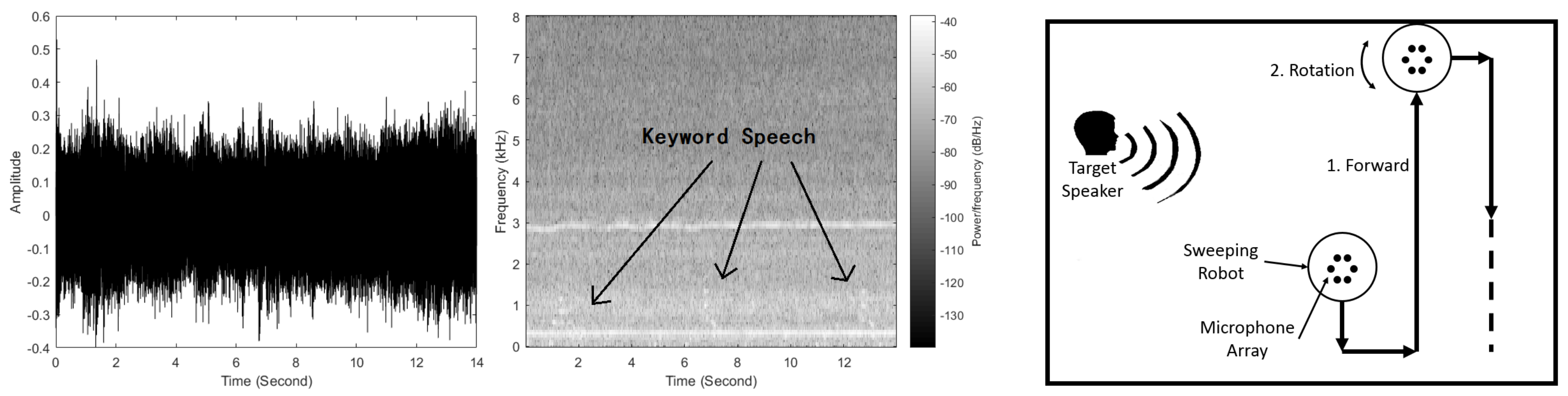
icassp 2022 Singapore

Alibaba Group 阿里巴巴集团

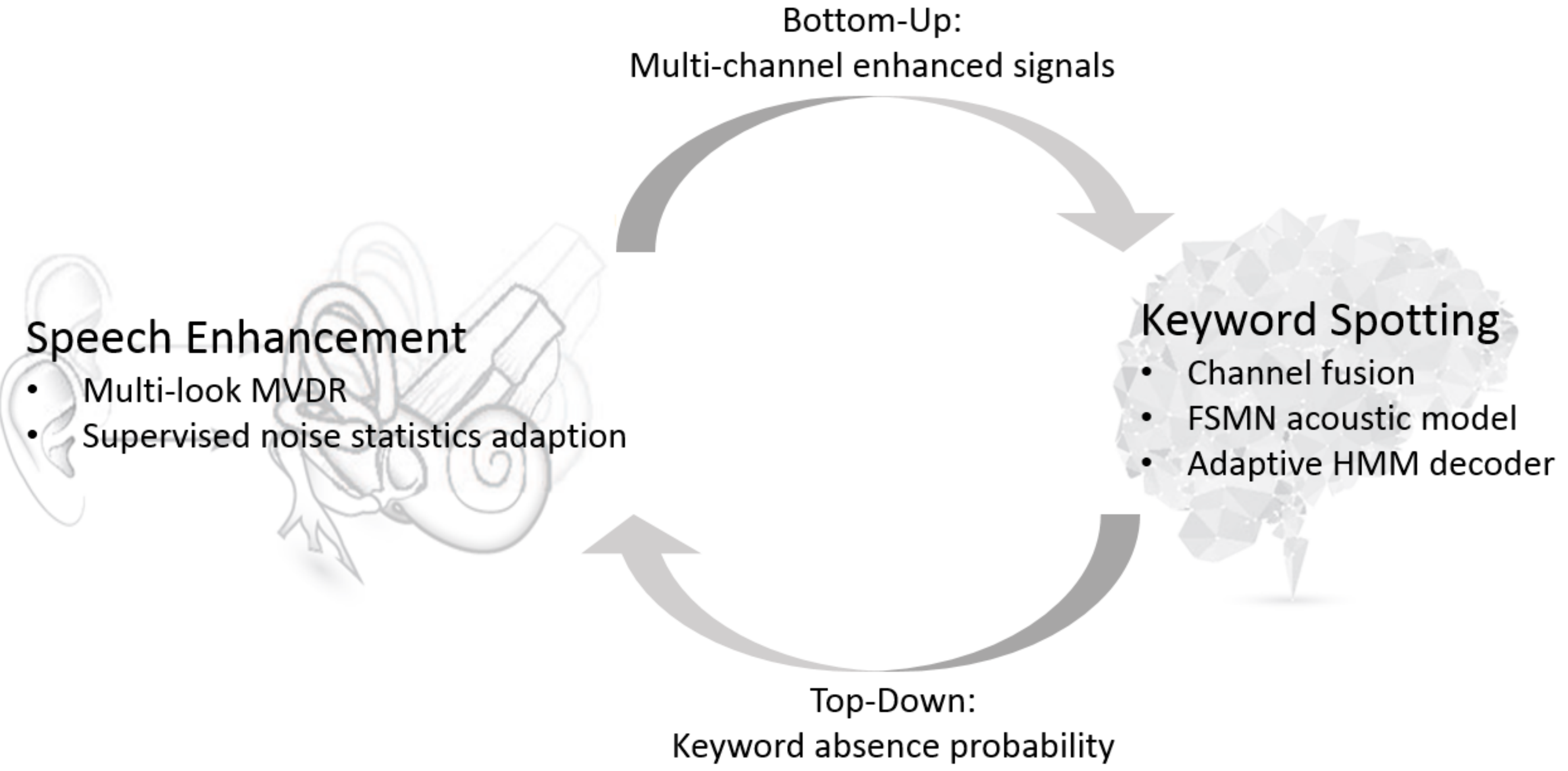
Paper ID: 1460

Introduction

The ego-noise (self-created noise) with complicated compositions, low SNR (-10 to -15dB), non-sparseness, and time variant target transfer function prevents the directly usage of audio interfaces on autonomous systems, such as sweeping robots.



A joint approach for ego-noise suppression and keyword spotting is introduced, 1. a feedback is established from keyword spotting to speech enhancement to perform better noise suppression, 2. an online algorithm is used to update the HMM decoder’s transition matrix.


Bottom-Up:
Multi-channel enhanced signals

Speech Enhancement
- Multi-look MVDR
- Supervised noise statistics adaption

Keyword Spotting
- Channel fusion
- FSMN acoustic model
- Adaptive HMM decoder

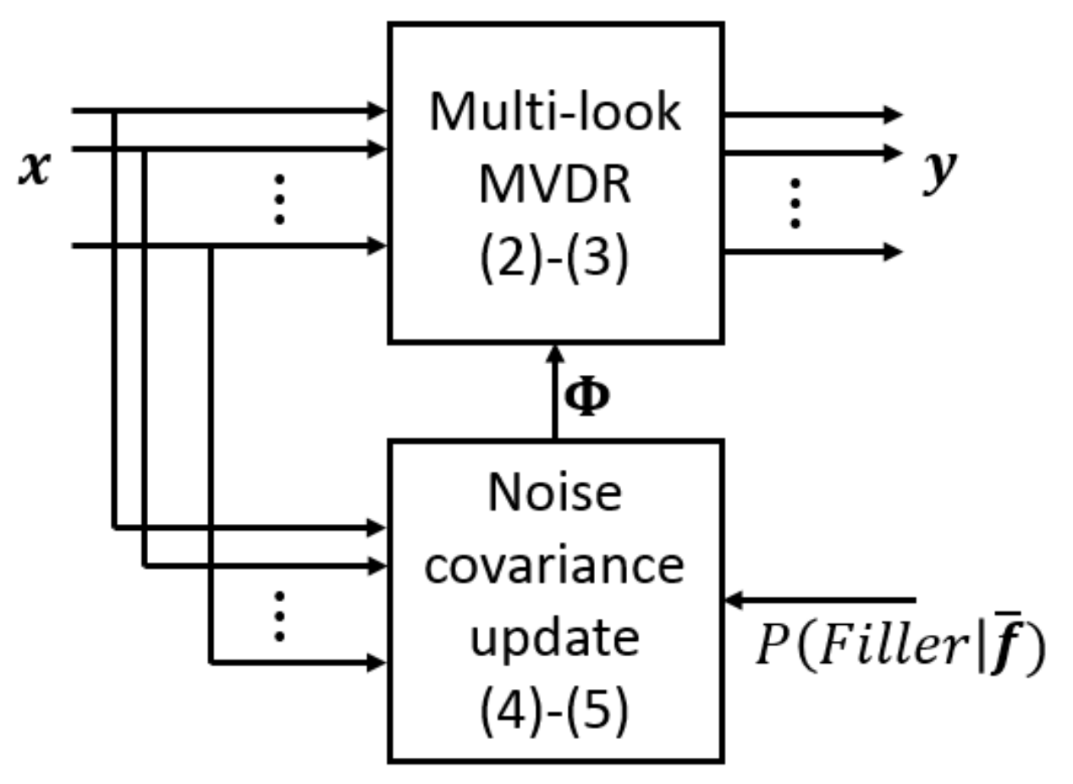
Top-Down:
Keyword absence probability

This system works more like the human auditory system, with both bottom-up and top-down processing between its sensory and cognitive subsystems [14].

[14] Huang, Yiteng, et al. "Supervised noise reduction for multichannel keyword spotting." *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

Speech Enhancement

Multi-look MVDR beamformers are generated from uniformly sampled look directions. To avoid speech distortion, the noise covariance matrix adaptation is slowed down when the keyword is present.


Multi-look MVDR (2)-(3)

Noise covariance update (4)-(5)

$$\boldsymbol{w}_n(\tau) = \frac{[\boldsymbol{\Phi}(\tau)+\epsilon\boldsymbol{I}]^{-1}\boldsymbol{d}(\varphi_n,\theta)}{\boldsymbol{d}^H(\varphi_n,\theta)[\boldsymbol{\Phi}(\tau)+\epsilon\boldsymbol{I}]^{-1}\boldsymbol{d}(\varphi_n,\theta)+\delta} \quad (2)$$

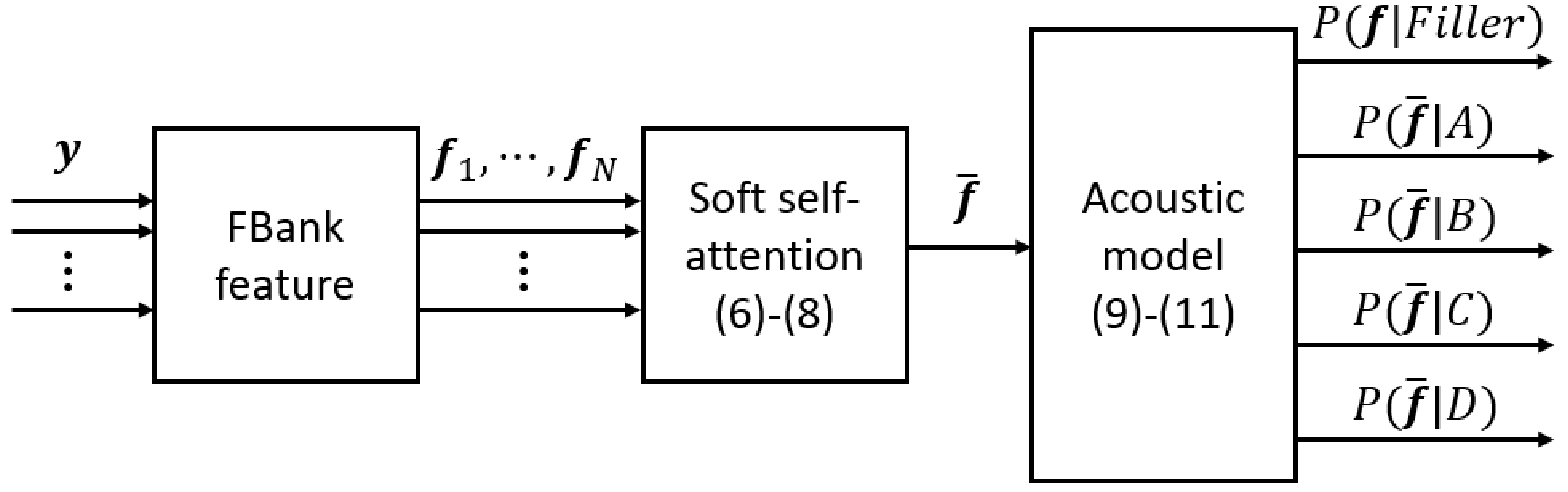
$$y_n(\tau) = \boldsymbol{w}_n^H(\tau)\boldsymbol{x}(\tau) \quad (3)$$

$$\boldsymbol{\Phi}(\tau) = \eta\boldsymbol{\Phi}(\tau-1) + (1-\eta)\boldsymbol{x}(\tau)\boldsymbol{x}^H(\tau) \quad (4)$$

$$\eta = \eta^{(1)} + (1-\eta^{(1)})(1-P(Filler|\bar{\boldsymbol{f}})) \quad (5)$$

Acoustic Model

Multi-channel features are fused by soft self-attention weights, feedforward sequential memory networks (FSMN) [18] is used as the acoustic model, which predicts the observation probabilities of 5 states: one filler, and four keyword characters: A, B, C, and D.


FBank feature

Soft self-attention (6)-(8)

Acoustic model (9)-(11)
$\boldsymbol{y}$
$\boldsymbol{f}_1,\cdots,\boldsymbol{f}_N$
$\bar{\boldsymbol{f}}$
$P(\bar{\boldsymbol{f}}|Filler)$
$P(\bar{\boldsymbol{f}}|A)$
$P(\bar{\boldsymbol{f}}|B)$
$P(\bar{\boldsymbol{f}}|C)$
$P(\bar{\boldsymbol{f}}|D)$

$$\bar{\boldsymbol{f}}(\tau) = \sum_{n=1}^{N}\bar{g}_n(\tau)\boldsymbol{f}_n(\tau) \quad (6)$$

$$g_n(\tau) = \boldsymbol{u}^T\tanh(\mathrm{Affine}(\boldsymbol{f}_n(\tau))) \quad (7)$$

$$\bar{g}_n(\tau) = \frac{\exp(g_n(\tau))}{\sum_{i=1}^{N}\exp(g_i(\tau))} \quad (8)$$

$$\boldsymbol{h}_0(\tau) = \mathrm{ReLU}(\mathrm{Affine}(\bar{\boldsymbol{f}}(\tau))) \quad (9)$$

$$\boldsymbol{h}_l(\tau) = \mathrm{FSMN}(\boldsymbol{h}_{l-1}(\tau)), l = 1, ..., L \quad (10)$$

$$\boldsymbol{p}(\tau) = \mathrm{Softmax}(\mathrm{Affine}(\boldsymbol{h}_L(\tau))) \quad (11)$$

[18] Zhang, Shiliang, et al. "Feedforward sequential memory networks: A new structure to learn long-term dependency." *arXiv preprint arXiv:1512.08301* (2015).

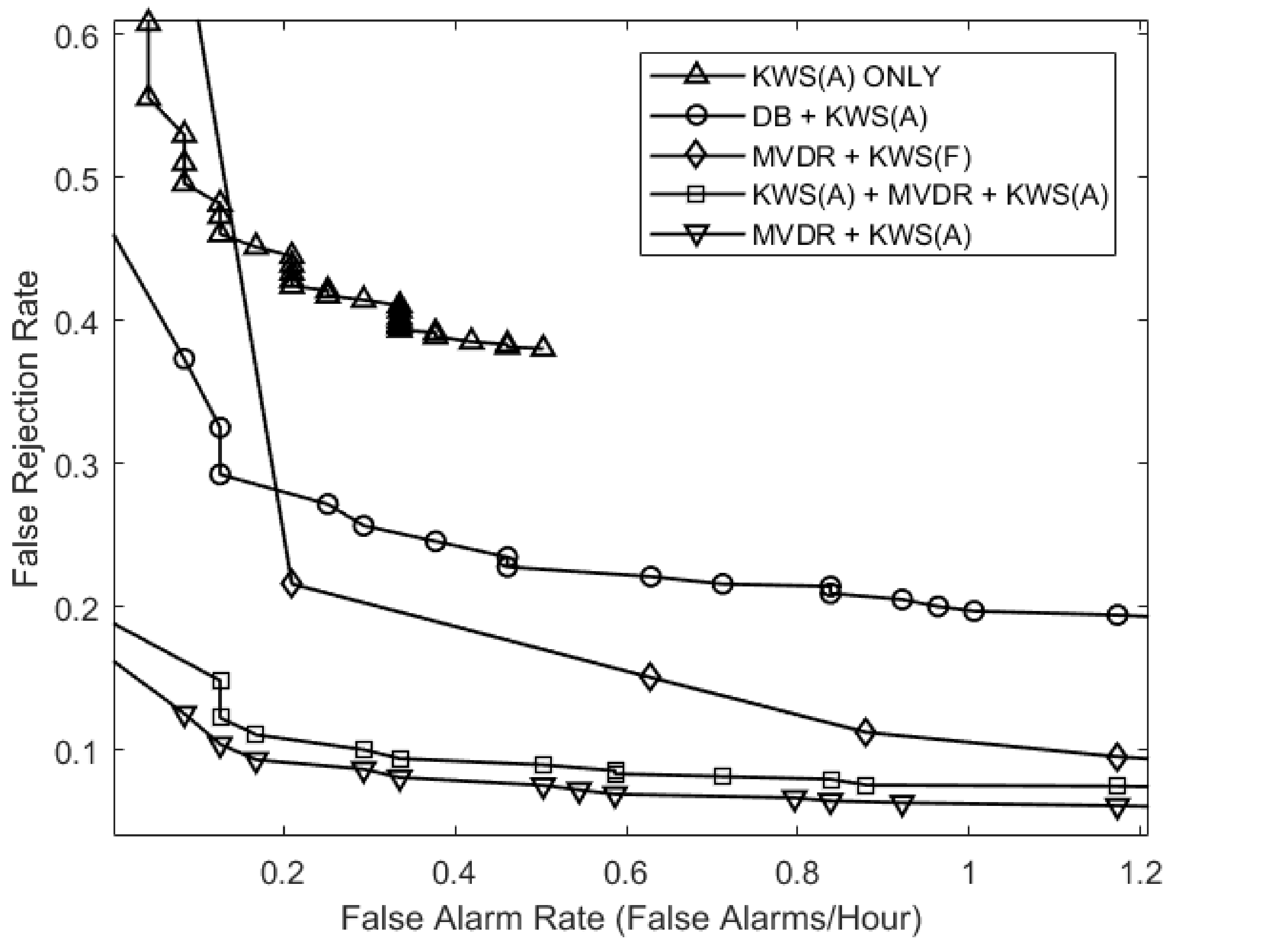
Decoder

A HMM decoder is used to smooth the acoustic model's outputs, the HMM's transition matrix is updated under the maximum likelihood criterion, which brings more flexibility in different noise conditions.

```
Algorithm 1: Online HMM transition matrix update.
  Initialize: $\boldsymbol{\alpha}(0), \boldsymbol{T}(0)$ empirically
  Input: $\boldsymbol{\alpha}(\tau-1), \boldsymbol{T}(\tau-1), \boldsymbol{p}(\tau)$
  Output: $\boldsymbol{\alpha}(\tau), \boldsymbol{T}(\tau)$
  1. Update each $\gamma_{ij}$ in $\boldsymbol{\Gamma}$ and each $\alpha_j$ in $\boldsymbol{\alpha}$:
        $\gamma_{ij}(\tau) = \alpha_i(\tau-1)t_{ij}(\tau-1)p_j(\tau)$   (12)
        $\alpha_j(\tau) = \sum_{i=1}^{5}\gamma_{ij}(\tau)$   (13)
  Then normalize:
        $\boldsymbol{\Gamma}(\tau) = \frac{\boldsymbol{\Gamma}(\tau)}{\sum_{i=1}^{5}\sum_{j=1}^{5}\gamma_{ij}(\tau)}$   (14)
        $\boldsymbol{\alpha}(\tau) = \frac{\boldsymbol{\alpha}(\tau)}{\sum_{i=1}^{5}\alpha_i(\tau)}$   (15)
  2. Update each $t_{ij}$ in $\boldsymbol{T}$, where $\eta^{(2)}$ is the fixed
  forgetting factor for HMM.
        $\eta_i = \eta^{(2)} + (1-\alpha_i(\tau))(1-\eta^{(2)})$   (16)
        $t_{ij}(\tau) = \eta_i t_{ij}(\tau-1) + (1-\eta_i)\gamma_{ij}(\tau)/\alpha_i(\tau)$   (17)
  Then, normalize $\boldsymbol{T}$ to row stochastic form.
```

Experiments

Experiments are carried out on real-world datasets, performances are visualized as ROC curves. The proposed approach (MVDR + KWS(A)) has better performance than fixed HMM decoder (MVDR + KWS(F)), and approaches without feed back (DB + KWS(A), KWS(A) + MVDR + KWS(A)).



The code for differential beamforming simulation is available: https://github.com/nay0648/ego2022