FP7-ICT-2013-C TWO!EARS Project 618075

Deliverable D4.2

Specification of feedback loops and implementation progress

Blauert, J., Brown, G., Bustamante, G., Cohen–L'hyver, B., Danès, P., Deshpande, N., Kim, Ch., Kohlrausch, A., Ma, N., Mohr, J., Obermayer, K., Pastore, T., Trowitzsch, I. & Walther, Th.*

November 27, 2015

* The TWO!EARS project (http://www.twoears.eu) has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 618075.

Project acronym:	Two!Ears
Project full title:	Reading the world with Two!Ears

Work package:	4
Document number:	D4.2
Document title:	Specification of feedback loops and implementation progress
Version:	1

Delivery date:	30th November 2015
Dissemination level:	Restricted
Nature:	Report

Editors:	Jens Blauert, RUB & Thomas Walther, RUB
Authors:	Armin Kohlrausch, Benjamin Cohen-L'hyver, Chungeun Kim, Gabriel Bustamante, Guy Brown, Ivo Trowitzsch, Jens Blauert, Johannes Mohr, Klaus Obermayer, Nikhil Deshpande, Ning Ma, Patrick Danès, Thomas Walther, Torben Pastore
Internal reviewers:	Klaus Obermayer

Contents

1 Introduction

The workplan of WP4 starts with the following statements.

> ‘Active listening entails bottom-up data processing as well as top-down mechanisms. To capture them, a framework must be set up that can host suitable feedback loops. It is the general task of WP4 to design an appropriate system architecture for this requirement, investigate meaningful feedback paths, implement them, and finally evaluate them regarding their functionalities. Input from other modalities than the auditory one will also be considered as a source of feedback information, particularly, position, direction and speed of head-&-torso movements (proprioceptive and sensorimotor input), and of identified optical objects (visual input)’.

In this context, the consortium has now made final decisions with regard to which possible feedback loops to investigate. Consequently, the process of implementation and evaluation of some of the selected loops has started. These activities belong to Task 4.2, the description of which reads as follows.

Task 4.2 Implementation of feedback loops

> ‘This is not an isolated task. It can only be performed in close interrelationship with WPs 1–3 & 5 (robotics component). Feedback loops will be designed and implemented following the strictly functional approach outlined on pp. 14/15 of the proposal and based on the in- and outputs of feedback loops as identified in Task 4.1. Implementation of feedback loops will be carried out stepwise, depending on state of development of affected system modules at various relevant stages of system. Stability problems will be investigated. During the implementation phase, the architecture of the Two!Ears system or at least of some of its modules will have to be suitably modified. Clearly, this process will last during most of project duration. Functional implications of the proposed feedback loops will be traced by analyzing responses to auditory and multi-modal stimuli similar to those used in the final evaluation in Task 4.4. Results will allow for grading feedback loops according to their functional relationships with possible brain mechanisms and thus provide means for their calibration – which would hardly be possible otherwise.’

It is obvious that the implementation of feedback loops depends very much on functional modules that have been set up and delivered by other workpackages. Consequently, the major effort of WP4 will be taken in the last project year. Nevertheless, relevant work is already going on, and some interesting results have already been achieved. Where necessary modules are not yet available for a fully fletched implementation on the Two!Ears core system, their function and output is simulated. To this end, a virtual testbed, the Bochum Virtual Test Environment (BEFT), is in the process of being developed. A lean version of it, (LVTE), which can already perform a number of relevant tasks, is described in Sec. 2.1 and was uploaded to the Git-repository. The full BEFT will be functional in due time. Nevertheless, it has to be stated that the results achieved so far with regard to auditory feedback are still fragmentary. Actually, this does not come by surprise, since a full integration of the feedback algorithms into the Two!Ears core system becomes only feasible once all necessary modules are at hand. This will be the case during the next months.

In the further course of this document, the sequence of Sections is structured according to a recent list of feedback loops to be considered and explored in Two!Ears. This list has been approved by the Project Board at the General Project Meeting in Toulouse, France, September 16–18, 2015. In the title of each Section and Subsection of the current document, reference is given to the respective feedback item in the following list.

Feedback loops to be considered and explored

- **(A)** Olivo-cochlear reflex (MOCR)
  - **(a1)** Unilateral, contralateral and central control – *RUB with TU/e, to be implemented for future experimentation only*
- **(B)** Insertion of supplementary signal-processing units, triggered by decisions based on information taken from the blackboard
  - **(b1)** Specific enhancement filters, such as for male voice, female voice, baby voice – *RUB with USFD and DTU*
  - **(b2)** Precedence-effect processor – *RUB with RPI, TU/e and DTU*
  - **(b3)** HRIR deconvolution – *RUB with URO and DTU*
  - **(b4)** Dereverberation algorithm – *reserve item*

- **(b5)** Binaural noise-reduction algorithm – *reserve item*
- **(b6)** Machine-learned source identification: feedback-based selection of features and classifiers – *RUB with TUB*
- **(b7)** Sensorimotor-cue processing – *RUB with CNRS*

- **(C)** Cognitive-level feedback, for example, on the basis of labeled environmental maps as built from information taken from the blackboard and from experts
  - **(c1)** Interpretation of scenes and assigning meaning to their elements – *RUB with UPMC, USFD and TUB*
  - **(c2)** Formation of attention and attention-based control of feedback processes – *RUB with UPMC and USFD*
  - **(c3)** Performing quality judgments from the listeners' point of view, based on internal references – *RUB with TUB*
  - **(c4)** Initiating robot maneuvers for scene exploration, for example, for object–distance determination, approaching sources, triangulation – *RUB with CNRS*
  - **(c5)** Keyword spotting – *RUB*
  - **(c6)** Requesting visual assistance through visual object localization and identification – *RUB with CNRS*

task	section/subsection
a1	2.8.1
b1	2.7, 2.11
b2	2.8.2, 2.11
b3	2.11
b4	–
b5	–
b6	2.9
b7	2.10
c1	2, 2.3, 2.4, 2.7
c2	2.7
c3	2.12
c4	2, 2.2, 2.4
c5	–
c6	2, 2.3, 2.4, 2.13

Table 1.1: Coverage of the individual feedback loops considered by the sections and subsections of the current deliverable D4.2

2 Integrating selected feedback paths into the TWO!EARS-system architecture

(The following relates to c1, c4, and c6)

The current expansion stage of the Two!Ears framework incorporates a plain blackboard system (cf. D3.1), which is intended to constitute the basis for feedback mechanisms of low to moderate complexity. To that end, the current blackboard structure had to be enhanced as to allow for multi-modal-feedback techniques and active exploration approaches. Herein, the construction of new knowledge sources becomes mandatory as well as the modification of existing expert subsystems.

In addition, a 'lightweight' component was developed to complement the *Bochum Experimental Feedback Testbed* (BEFT) software package. This new component, *Lean Virtual Test Environment* (LVTE), enables quick and reliable testing of basic feedback routines. It integrates the *Sound-scape Renderer* (SSR) for auralization, and is enabled to communicate seamlessly with the current blackboard architecture.

Note that the methods and algorithms proposed in Secs. 2.1, 2.2, & 2.3 are fully integrated in the Two!Ears framework and are available on the project's internal repository. However, further in-depth testing will be necessary prior to public dissemination

The section below focuses on a comprehensive description of the LVTE and is followed by in-depth analysis of the knowledge sources that were set up to enable basic multi-modal feedback within the LVTE/blackboard system.

2.1 The Lean Virtual Test Environment (LVTE)

Intended as a direct descendant of the more powerful BEFT, LVTE avoids the overhead of the former system by stepping away from high-end visual simulation. Instead, visual stimuli are generated artificially through degradation of environmental ground-truth information, following the proposals found in [43]. LVTE is intended to quickly perform experiments with multi-modal feedback methods and active exploration procedures. The system is solely based on MATLAB®, thus ensuring seamless integration into the Two!Ears project framework. Note that the LVTE can readily be superseded by the fully fletched BVTE or the physical robot, once the assessed routines become sufficiently stable.

2.1.1 System overview

Taking on the role of the *robotConnect* interface defined in the blackboard architecture, the LVTE emulates the robotic front-end, thereby allowing dedicated knowledge sources to initiate platform/head motion of the virtual robot. Moreover, the knowledge sources are enabled to poll environmental data via the *robotConnect* interface. Such data is mandatory for informed hypotheses generation and decision making. To achieve synchronization between LVTE and the blackboard system, the *UpdateEnvironment* knowledge source – see below – was introduced.

Directly interfacing with the SSR, LVTE allows for generating the virtual robot's ear signals and sends them to the blackboard system for further processing by the *AuditoryFrontEnd* knowledge source. For better comprehensibility, Fig. 2.1 graphically subsumes the overall structure of the proposed LVTE/blackboard system.

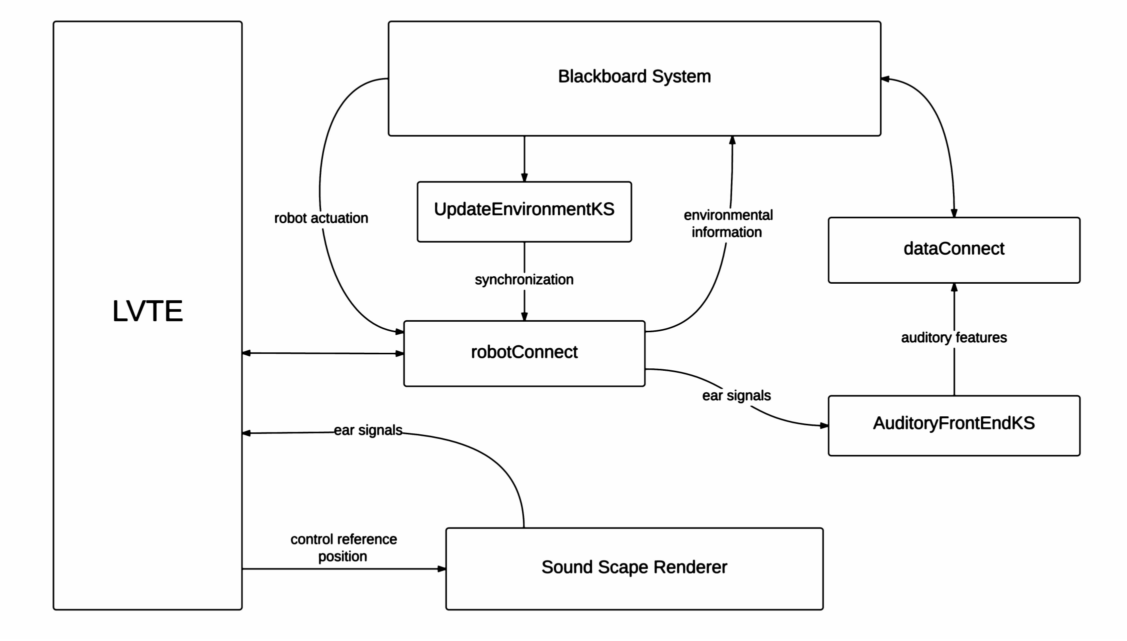


Figure 2.1: Integration of the LVTE into the current blackboard architecture

The system architecture as depicted in the figure was completely set up using the MATLAB® programming environment, thus ensuring compatibility across platforms and with the generic TWO!EARS framework.

2.1.2 Class structure

The following discussion re-assesses the system sketched in Fig. 2.1, getting granular on the LVTE block. This block consists of multiple MATLAB® classes, which are analyzed in detail below. Fig. 2.2 augments the following analysis by giving a comprehensive graphical overview of LVTE's class structure.

'Environment' class

The *Environment* class constitutes the basis of the LVTE system. It allows the programmatic definition of arbitrary scenarios of low to medium complexity. The scenarios are standardly situated in a *shoe-box* environment. The experimenter is enabled to define room dimensions manually and to freely place the virtual robot within this room. The scenario duration can be chosen within an interval of $[0 \ldots T_S]$.

Note that, currently, the SSR stops the auralization process as soon as there are no active sound sources. Such a 'silent' condition will, however, often occur in upcoming experiments and would cause early termination of the simulation. To counter this issue, an auxiliary 'silent' source was introduced. This muted sound source generates no audible data, yet remains active for $[T_S]$ and ensures correct auralization of the complete scenario.

To accommodate for the cognitive ideas pursued in TWO!EARS, auditory stimuli used in the auralization process are always related to a dedicated *auditory category* that is derived from the IEEE sound database *AASP* – see [39]. The initial *DASA*-related set of auditory categories is defined as $\mathcal{S}_A = \{\textbf{`speech'}, \textbf{`alert'}, \textbf{`knock'}\}$. Note that for the testing done here, sound samples from the corpus were used regardless of whether or not they had been employed for classifier training. This method is a plain 'proof-of-concept' and has to be re-assessed in upcoming system tests, using a wider variety of test data. Assume that $\mathcal{S}_A\,[j]$ allows to access the j^{th} element of the category set, a shortcut, $C_j^A = \mathcal{S}_A[j]$, then extracts the j^{th} category from the set and facilitates further notation.

For each category in $\mathcal{S}_A$, a subset of concrete *stimuli instances* can be defined, for instance, $\mathcal{I}_{\textbf{`speech'}} = \{\text{`speech01.wav'}, \text{`speech02.wav'}, \ldots\}$. This set of stimuli/instances will be enlarged depending on the availability of appropriate sound data. In addition to the auditory categories/instances, the *Environment* class stores a set of *visual categories*,

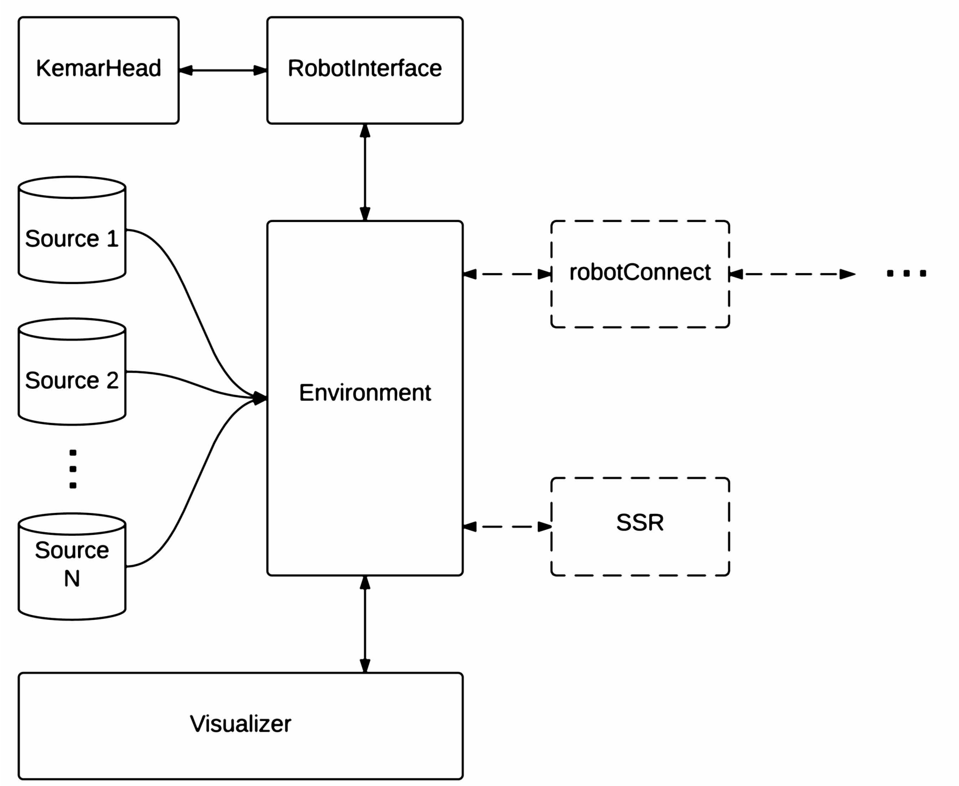


Figure 2.2: The LVTE class hierarchy in detail. The blackboard system is omitted for visual clarity and indicated by the three dots on the right hand side

$\mathcal{S}_V$, as become mandatory in audio-visual experiments. Matching $\mathcal{S}_A$ cognitively, the set of visual categories is currently defined as $\mathcal{S}_V = \{\textbf{`person'}, \textbf{`siren'}, \textbf{`door'}\}$ and will grow according to upcoming extensions of $\mathcal{S}_A$. As in the auditory case, $\mathcal{S}_V\,[j]$ allows to access the j^{th} element of the visual category set. Again, the shortcut $C_j^V = \mathcal{S}_V[j]$ holds.

Adding up to the above, *Environment* maintains a set of audio-visual sources that can be arbitrarily placed by the experimenter. Data from each inserted source is automatically transferred to the SSR, thus transparently generating the auditory scene. Each audio-visual source is represented by an instance of the class *Source*. This class is assessed below in more detail.

'Source' class

The *Source* class stores basic information concerning all defined audio-visual sources. Mind that source locations are memorized together with the sources' names and the acoustic stimuli emitted. The latter can be altered at runtime, enabling simulation of sources with multiple utterances. Each audio-visual source, i, maintains a time-line, $\mathcal{T}_i$, controlling the onset and offset of stimulus instances chosen from $\mathcal{I}_{C_i^A(t)}$, where $C_i^A(t)$ is the auditory category of source i at time t.

This *auditory schedule* is used by the *Environment* class to control SRR-based auralization in arbitrarily complex scenes. Note that auditory schedules can either be generated manually or can be randomly created using task-specific setup routines. In addition to the auditory schedule, each source comprises a *visual schedule*. This is useful for audio-visual experiments in the virtual environment.

Visual scheduling allows for freely assigning all categories in $\mathcal{S}_V$ to the constructed sources. Note that the visual schedule currently does not enable the changing of any assigned category at runtime. This feature will later be added – if deemed necessary for upcoming experiments.

In addition to the list of audio-visual sources, the *Environment* class comprises a *RobotController* interface class that acts as a proxy for the virtual robot. The following paragraph focuses on the description of this interface class.

'RobotController' class

Mimicking the robotic front-end, the *RobotController* class is intended to emulate important properties/capabilities of the physical device and to provide the LVTE with access to these features. In its current construction stage, the *RobotController* entity simply maintains the position of the robot and contains an interface to the virtual *KEMAR* head, which is mounted on top of the simulated robotic platform.

Changes in the robot's position are directly propagated to the SSR, in order to synchronize the auralization with current scenario conditions. In forthcoming versions of the system, the *RobotController* class will integrate information from laser scanners or other sensors that provide environmental information. Further, the class interface is readily expandable to handle kinematics that exceed pure head rotation and platform translation.

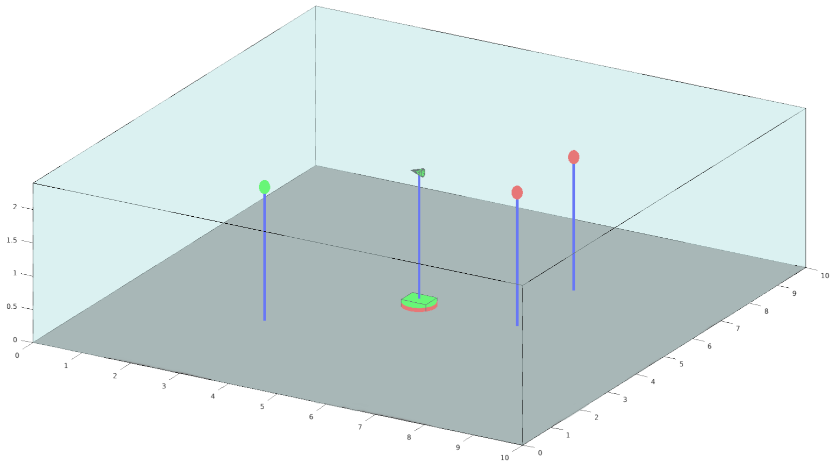


Figure 2.3: 3–D visualization of a standard scenario. The green cone indicates the robot's head, with the cone's tip indicating the looking direction. The green box represents the robotic platform. Multi-modal sources are depicted as spheres. Red means that the source is muted. Green corresponds to an active source. The dimension of the sketched shoe-box room is 10x10x2.4 m

'KemarHead' class

Within the *KemarHead* class, the LVTE provides all functionality required to control the rotation of the artificial Kemar head attached to the simulated robotic platform. The class also maintains information concerning the HRTFs used for auralization by the Sound-Scape Renderer (SSR). Currently, the virtual head is equipped with transfer functions as recorded at TUB (azimuth resolution: 1 °, distance: 3 m). Other HRTF datasets can be utilized if necessary for future experiments.

'Visualizer' class

To visualize the simulation of moderately complex audio-visual scenarios in shoebox-room geometries, the LVTE comprises a dedicated *Visualizer* class, which provides the experimenter with a simplified 3–D view of the current scenario status. As stated above, visualization in the LVTE is not geared towards high visual fidelity and speed. Instead, cross-platform compatibility and ease of use are prioritized.

To that end, the 3–D visualization is completely based on MATLAB® routines, thus avoiding the need for separate installation of bulky 3–D engines (such as OGRE [28]). Despite their simplicity, the MATLAB®-based scenario sketches allow well to visually track the robot's behavior at runtime – as Fig. 2.3 demonstrates.

2.2 Multi-modal feedback – forming audio-visual objects

With the LVTE block beeing analyzed, the focus of the following discussion now turns to the *Blackboard System* (cf. Fig. 2.1). Speaking of the *Blackboard System*, please note that this term comprises the blackboard core elements (including event managing and scheduling) and all bound knowledge sources. The blackboard core has currently been left unaltered (see D3.1 for details of the blackboard core structure).

The blackboard system has to seamlessly interact with the LVTE in order to acquire audio-visual data and to provide basic attention/feedback mechanisms. To achieve these goals, the creation/modification of knowledge sources and the definition of purposeful 'binding schemes' become mandatory.

The following overview concentrates on the knowledge sources. A basic, yet fully functional blackboard system was constructed that at least fulfills the *administrative chores* listed below.

- The system is able to step the LVTE simulation flow
- It is enabled to control the virtual robot in the LVTE
- Acquisition of simulated audio-visual data is supported
- The acquired data can be displayed on a per-time-step basis

In addition, the system has to solve a comparatively simple task in the multi-modal feedback domain, as sketched in the next paragraph.

2.2.1 Knowledge sources – definition and binding pattern for audio-visual object formation

Assume that three sources, $\mathbf{q}_{1...3}$, are set up at previously defined locations. These sources emit stimuli with a duration of 2 s in a sequential pattern, $\mathbf{q}_1 \rightarrow \mathbf{q}_2 \rightarrow \mathbf{q}_3$. Intervals between the single emissions last 1 s. The robot listens to the sources and forms an auditory-object hypothesis for each of the overheard stimuli. It then uses the interval between two consecutive emissions to turn into the direction of the formerly active source.

Eventually, it finds the visual category of the focused source and, consequently, forms an audio-visual-object hypothesis, namely, if and only if the estimated auditory and the observed visual category coincide. Otherwise, the initial auditory-object hypothesis is rejected and the experimenter will be informed. Below, the knowledge sources required to address the above challenge are investigated in detail.

UpdateEnvironmentKS

The *UpdateEnvironment* knowledge source descends from the *AbstractKS* defined in the current blackboard architecture. It constitutes the basis for communication with the LVTE, namely, by initiating time-stepping of the LVTE from within *UpdateEnvironmentKS*. This includes triggering the *Visualizer* mechanism that sketches the virtual environment at each time step.

UpdateEnvironmentKS is executed continuously and is directly bound to the blackboard scheduler. The knowledge source is triggered as soon as the blackboard agenda becomes empty. By updating the environment in a cyclic way, *UpdateEnvironmentKS* constitutes a necessary prerequisite for all knowledge sources that incorporate data from the LVTE system, such as the *VisualDisplayKS*, the *AuditoryDisplayKS* or the *VisualIdentity* knowledge source (see below).

The update schedule followed by *UpdateEnvironmentKS* directly depends on the block/chunk size, B, (given in signal samples) and the signal sampling frequency, f_S, used in the SSR. In order to ensure audio-visual synchrony, the LVTE is triggered by the aforementioned knowledge source every B/f_S. Note that current setting are: f_S=44100 Hz, B=2048 samples.

SignalLevelKS

This knowledge source descends from the *AuditoryFrontEndDepKS* and is responsible for detecting the signal level in a processed audio chunk. To that end, assume that $\mathbf{c}_i$ is an auralized sound chunk sampled at simulation cycle i with the standard sampling rate f_S. Then, the *variance* of this chunk equals

$$\sigma^2(\mathbf{c}_i) = \frac{1}{N}\sum_{n=1}^{N} |\mathbf{x}_n - \boldsymbol{\mu}_x|^2\,, \tag{2.1}$$

where $N = B$ is the number of samples in the chunk, and $\mathbf{x}_n$ is a value from $\mathbf{c}$, sampled at $t = \frac{n}{f_S}$ s. The chunk variance represents the 'power of the (partial) signal with its mean removed' [27]. Mind that this measure is only one option to express the signal level. The *total energy* or the *average power* of the chunk could be used as well – compare [27].

Note that the signal level provides a convenient measure to assess (on a per-chunk-basis) the reliability of the auditory classifiers currently used in the Two!Ears framework. This renders the *SignalLevelKS* a mandatory precursor for several of the knowledge sources discussed below.

LocalizerKS

LocalizerKS is nearly identical to *LocationKS* employed in the current blackboard architecture. However, *LocalizerKS*, contrary to *LocationKS*, is called in each simulation cycle. Further, *LocalizerKS* is triggered by observing results posted by *SignalLevelKS*, namely, when $\sigma^2(\mathbf{c}_i) > \epsilon$ holds, then the former KS estimates the azimuth of the incoming signal. Currently, ϵ is set to the limits of machine precision.

AuditoryIdentityKS

The auditory-identity knowledge sources engaged in the actual context also descend from *AuditoryFrontEndDepKS* and are largely identical to *IdentityKS*-code fragments found in the current Two!Ears architecture.

Going beyond the original formulation, *AuditoryIdentityKS* employs an energy criterion for deciding whether or not to trust the results of its internal sound classifier. Actually, an *auditory identity hypothesis,* $\mathcal{H}^a$, as generated by this KS is trusted if and only if $\sigma^2(\mathbf{c}) > \epsilon$. Otherwise, the probability of $\mathcal{H}^a$ is clamped to zero. This practice avoids the processing of false positive hypotheses in signal intervals with no audible data.

Since the LVTE/blackboard system has to function in a multi-modal environment, the knowledge source generates an *auditory-identity hypothesis* (see above) in order to distinguish its estimates from *visual-identity hypotheses* as generated by *VisualIdentityKS* – see below.

Note that the proposed system currently comprises three different auditory knowledge sources, that is, *SpeechIdentityKS*, *KnockIdentityKS*, and *AlertIdentityKS*. So far, these entities have to be called in a daisy-chain manner. This issue remains to be addressed in later system versions.

VisualIdentityKS

Complementing the auditory identity knowledge sources, *VisualIdentityKS* was created from scratch – inheriting from *AbstractKS* in order to capture visual-identity information from the current scenario. To that end, the knowledge source re-uses the approach found in [43] as follows. Let $\mathcal{Q} = [\mathbf{q}_1, ..., \mathbf{q}_{N_Q}]$ be the set of all sources in the LVTE. Further, let $\mathbf{p}_i$ define the position of audio-visual source i in the azimuthal plane. Be C_i^V the true visual category of source i, that is, this category stems from the set $\mathcal{S}_V$. In addition, define $\mathbf{r}$ as the current position of the robot.

Now, let $\mathbf{d}_i = \mathbf{p}_i - \mathbf{r}$ be a vector that extends from the robot's center to the center of source i. With that, set up the distance, $d_i = ||\mathbf{d}_i||$, from the robot to source i. Further, let $\mathbf{h}_r$ be the current looking direction (that is, the heading vector) of the robot. Define

$$\phi_i = \arccos\left(\frac{\mathbf{h}_r \cdot \mathbf{d}_i}{||\mathbf{h}_r|| \cdot ||\mathbf{d}_i||}\right) \tag{2.2}$$

to be the relative azimuth between the robot's looking direction and source i. Based on the above definitions, let

$$D_d(d_i) = 1 - \frac{1}{1 + e^{\frac{d_i - 10}{2}}} \tag{2.3}$$

represent a *degradation function* that relates the leveling of visual sensor reliability to an increasing distance between the robot and source, i. Let further

$$D_a(\phi_i) = e^{-0.5(\frac{\phi_i}{90})^2} \tag{2.4}$$

constitute a degradation function that sketches the loss in sensor reliability caused by an increase in the relative azimuth between the robot and source i. Unifying the above degradation functions, let

$$v_{P,i} = v_{P,i}(d_i, \phi_i) = D_d(d_i) \cdot D_a(\phi_i) \tag{2.5}$$

be the *visual perceptibility* of source i – compare [43]. Let further C_j^V sample all visual categories in $\mathcal{S}_V$. With that, a bifurcating function

$$p_i(C_j^V, v_{P,i}) = \begin{cases} 0.5 + \frac{0.5}{e^{20 \cdot \left(v_{P,i} - 0.5\right)}}, & \text{if } C_i^V = C_j^V \\ 0.5 - \frac{0.5}{e^{20 \cdot \left(v_{P,i} - 0.5\right)}}, & \text{if } C_i^V \neq C_j^V \end{cases} \tag{2.6}$$

can be set up that computes the *category membership* estimates for source i as follows. If the robot is spatially close to the source and looks directly towards it, the estimate for the source's true category, $C_j^V = C_i^V$, approaches 1.0, whereas the estimates for all other categories, $C_j^v \neq C_i^V$, tend to zero. If the device turns/steps away from source i, the estimates for all categories, $p_i(C_j^V, v_{P,i})$, settle around 0.5, thus becoming equally distributed.

Equation 2.6 intuitively approximates the information content that can reasonably be expected from simulated or physical vision sensors. Note that the above list of degradation functions can easily be extended. It would, for instance, be straightforward to define sensor degradation caused by diminished illumination or by fog.

With the above, the visual-identity knowledge source computes the observation probability, $p(C_k^V)$, for a dedicated visual category with index k as follows,

$$p(C_k^V) = \arg\max_i p_i(C_k^V, v_{P,i})\,. \tag{2.7}$$

The observation probabilities for all visual categories are eventually stacked into a vector and pushed to the blackboard. Mind that the visual-identity knowledge source generates all category hypotheses simultaneously. Therefore this KS needs to be called only once per time step in the blackboard-scheduling scheme, contrary to the auditory-identity knowledge sources, which have to be fired sequentially.

ReactToStimulusKS

Inheriting from *AbstractKS*, *ReactToStimulusKS* allows to form *auditory-object hypotheses* in the following way. The knowledge source monitors the level of the audio signal as provided by *SignalLevelKS*. Assume that $\sigma^2(\mathbf{c}_i)$ exceeds the trigger threshold, ϵ, in simulation cycle i_{on} and remains above the threshold until cycle i_{off}. Assume further that exactly one audio-visual source, $\mathbf{q}_s$, can be active in each simulation cycle. Consequently, source activation is postulated to be sequential.

With that, it can reasonably be expected that the sound data sampled in the interval $I_{act} = [i_{on} \dots i_{off}]$ corresponds to exactly one stimulus emitted by source $\mathbf{q}_s$. Following this reasoning, *ReactToStimulusKS* accumulates auditory information within the aforementioned interval in order to formulate a hypothesis concerning the location and identity of the perceived stimulus. To that end, the azimuth posterior distribution, $p_i(\phi)$, computed by *LocalizerKS*, is retrieved at each $i \in I_{act}$ and added to the source's *location accumulator*, $p_s(\phi)$, according to

$$p_s(\phi) = \frac{1}{|I_{act}|} \sum_{n=i_{on}}^{i_{off}} p_i(\phi)\,. \tag{2.8}$$

Further, identity information is acquired from all available *AuditoryIdentityKS* as follows. Assume there exist auditory-identity-knowledge sources, $K^1_{Aud.} \dots K^{|\mathcal{S}_A|}_{Aud.}$. Let $\mathcal{H}^a_{j,i} = \mathcal{H}(K^j_{Aud.})$ be the auditory-identity hypothesis computed by $K^j_{Aud.}$ in simulation cycle i. Then, $\mathcal{H}^a_{j,i}$ reflects the probability for the overheard stimulus at cycle i to belong to auditory category j. From that the average category membership for the stimulus emitted by source $\mathbf{q}_s$ can be synthesized as follows,

$$\mathcal{H}_j^a(\mathbf{q}_s) = \frac{1}{|I_{act}|} \sum_{n=i_{on}}^{i_{off}} \left(\mathcal{H}_{j,i}^a\right)^3 . \tag{2.9}$$

Note that the cubic extension in the above formula was chosen empirically in order to assign higher weight to sharply peaked probability distributions.

With the above, let

$$\phi_s = \arg\max_{\phi} p_s(\phi), \text{ with } \phi \in [0..2\pi] , \tag{2.10}$$

$$K_s^A = \mathcal{S}_A \left[\arg\max_{j} \mathcal{H}_j^a(\mathbf{q}_s) \right] , \text{ with } j \in [1, ..., |\mathcal{S}_A|] \tag{2.11}$$

be the putative azimuth of $\mathbf{q}_s$, respectively, the source's expected auditory category label. Given ϕ_s and K_s^A, *ReactToStimulusKS* forms an *auditory-object hypothesis,* $\mathcal{H}_s^o = \{\phi_s, K_s^A\}$, for the active source and sends this hypothesis to the blackboard memory. Note that the integration method used above renders quite robust azimuth/category estimations. The results as generated in anechoic conditions appear to be reliable, even without integration of *ConfusionKS* or *ConfusionSolvingKS*.

TurnToKS

The *TurnToKS* descends from *AbstractKS* and reads from blackboard memory the auditory object hypothesis as generated by *ReactToStimulusKS*. Given the putative azimuth, ϕ_s, for an active sound source, $\mathbf{q}_s$, *TurnToKS* triggers robot rotation in the LVTE. This rotation is executed by the robot simulator in a completely transparent manner, which means that the blackboard is not blocked while the platform homes in on ϕ_s. Note that the simulator always computes the most time-efficient-rotation pattern, taking into account the maximum angular velocity of the robot device. Once ϕ_s is reached, the LVTE broadcasts a notification event, which is received by*BuildAudioVisualObjectKS* described below.

BuildAudioVisualObjectKS

As soon as the above *TurnToKS* focused a hypothetical acoustic source at ϕ_s, *BuildAudioVisualObjectKS* takes over and tries to fuse audio and visual information. To that end, the knowledge source reads the category estimate stored in $\mathcal{H}_s^o$ and compares it to the the most plausible visual category inferred from the results delivered by *VisualIdentityKS*. For the focused source $\mathbf{q}_s$, this category is found via

$$K_s^V = \mathcal{S}_V \left[\arg\max_k p\left(C_k^V\right) \right] . \tag{2.12}$$

To verify or falsify audio-visual category pairs, the blackboard is primed with a list of correct pairs, L^{ok}. For the current experiment, let

$$L^{ok} = \{(\text{‘}\mathit{person}\text{’}, \text{‘}\mathit{speech}\text{’}), (\text{‘}\mathit{door}\text{’}, \text{‘}\mathit{knock}\text{’}), (\text{‘}\mathit{siren}\text{’}, \text{‘}\mathit{alert}\text{’})\} . \tag{2.13}$$

If the pair $\{K_s^A, K_s^V\}$ is in L^{ok}, *BuildAudioVisualObjectKS* acknowledges correctness of the initial auditory hypothesis and builds a blackboard entry for a new audio-visual object, $\{K_s^A, K_s^V\}$, at $\phi_s\,^\circ$ azimuth. If $\{K_s^A, K_s^V\} \notin L^{ok}$, the initial auditory hypothesis is assumed to be wrong, causing the knowledge source to notify the experimenter and discard the inferred audio-visual object. At this point, low-level algorithms could be called to enhance auditory-feature extraction and correct the erroneous inference.

VisualDisplayKS and AuditoryDisplayKS

VisualDisplayKS and *AuditoryDisplayKS* enable the experimenter to continuously observe the hypotheses generated by *VisualIdentityKS* and the auditory-identity knowledge sources. The auditory-display knowledge source descends from *AuditoryFrontEndDepKS*. Its structure and purpose are inspired by *IdTruthPlotKS* found in the current Two!Ears framework.

In contrast, the visual-display knowledge source was constructed from scratch and inherits from *AbstractKS*. Note that neither *VisualDisplayKS* nor *AuditoryDisplayKS* are knowledge sources in the literal sense. They should instead be considered as auxiliary mechanisms that enable straightforward debugging in the LVTE/blackboard system.

It remains to set up a purposeful *binding scheme* for linking the knowledge sources listed above. Even for the comparatively straightforward task of audio-visual-object formation, the resulting connection pattern becomes quite complex – as Fig. 2.4 shows. Note that the activation pattern of the network is indicated by the arrow tips. An arrow pointing from knowledge source A to knowledge source B means that the latter is activated after the former. The depicted blackboard scheduling is experimental and is going to be re-assessed in upcoming system versions.

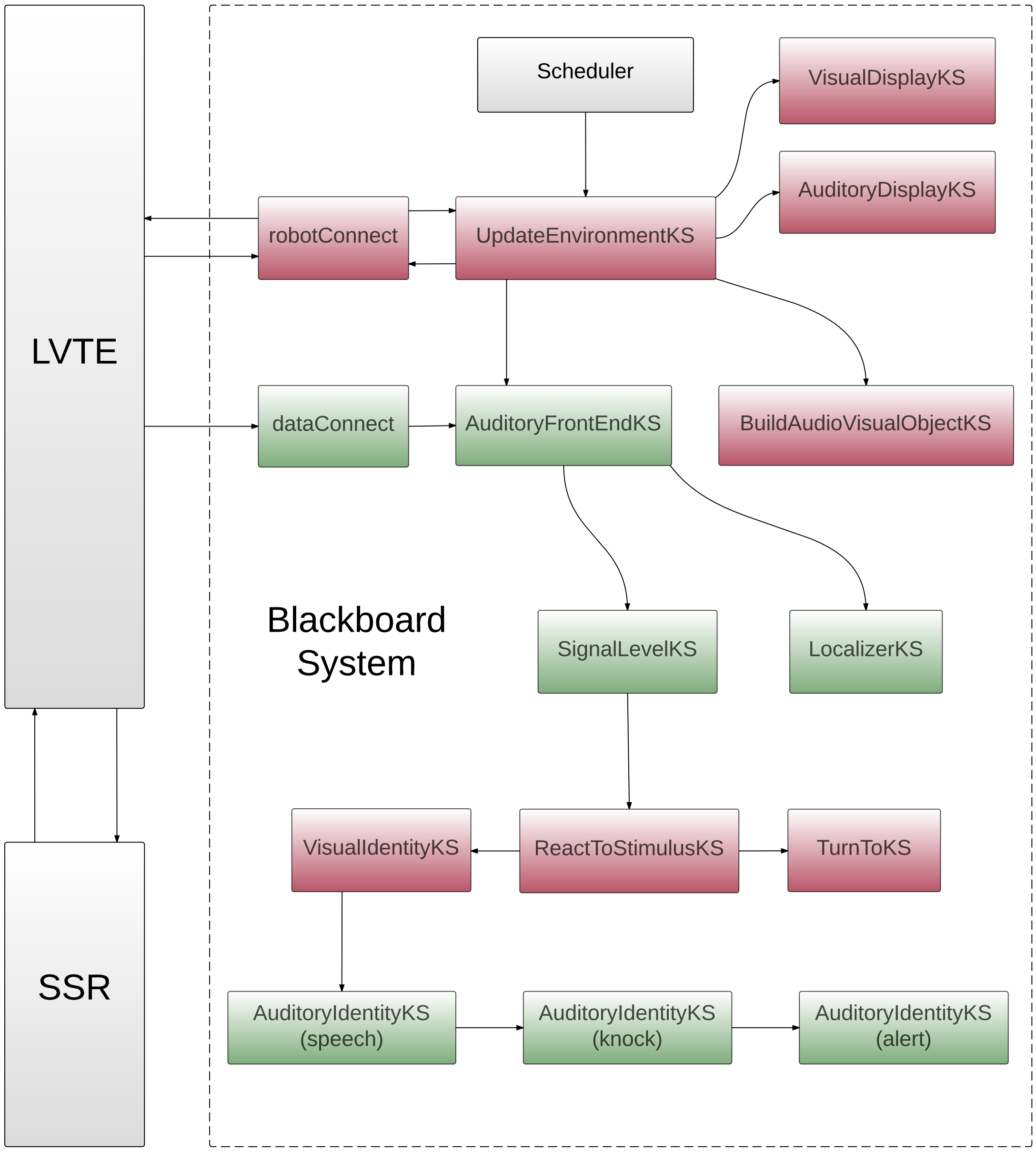


Figure 2.4: The basic LVTE/blackboard system configured for solving the audio-visual-object formation task (see running text). Colors indicate additional connectivity. Red means that the tinted KS can communicate with LVTE via the robotConnect variable of the blackboard. Green means that the corresponding KS is dependent on the blackboard's AFE connection

2.2.2 Experiments regarding audio-visual object formation

To assess the the above audio-visual-object-formation scheme with respect to localization precision and auditory categorization, plain test conditions were defined as follows. $N_T = 100$ scenarios (10 x 10 m, anechoic conditions) were auto-generated with a randomly placed single source, $\mathbf{q}_{test}$. Let $\mathbf{p}_{test}^{GT}$ represent the randomly chosen ground-truth coordinates of $\mathbf{q}_{test}$ in the x–y plane. Assume that the onset of source activity is at $t = 0\,\text{s}$, its offset at $t = 1.5\,\text{s}$. The stimulus emitted by $\mathbf{q}_{test}$ is randomly selected from IEEE's *AASP* categories $\{\textbf{‘speech’}, \textbf{‘alert’}, \textbf{‘knock’}\}$ as found in the Two!Ears data repository. The complete duration of the scenario is $T_D = 3\,\text{s}$. The robot is placed at $\mathbf{r} = [5, 5]^T$.

In each of the N_T scenarios, the task is to acquire the unknown stimulus, locate it by means of its global azimuth angle and form an auditory-object hypothesis. Then, the robot platform turns towards the estimated location of the putative auditory object and uses simulated visual information to verify the auditory-object hypothesis. If auditory and visual information coincide, the system forms a new audio-visual object and sends it to the blackboard.

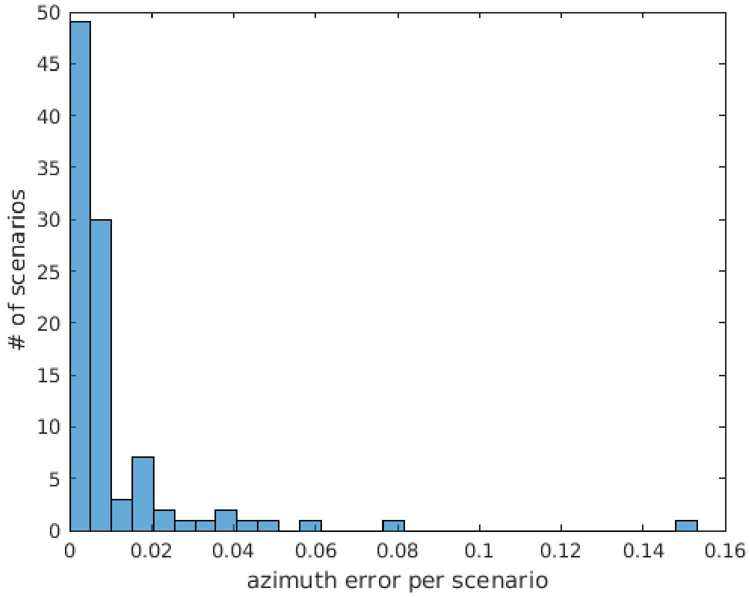


Figure 2.5: Experimental localization performance of the Two!Ears system in audio-visual object formation. The error in ϕ (azimuth) accumulates around 0.0103 rad (0.5889 °). See running text for details

If, however, the acquired visual information does not comply with the initial auditory-object hypothesis, the latter is falsified. Please note that with the above, performance measures can be set up for the proposed audio-visual-object-formation routines in the following way. First of all, *ReactToStimulusKS* should localize $\mathbf{q}_{test}$ as precisely as possible. To that end, assume that the estimated azimuth of $\mathbf{q}_{test}$ is ϕ^E_{test}. Then, define the vector $\mathbf{v}^E_{test}[\cos(\phi^E_{test}), \sin(\phi^E_{test})]^T$ that describes the direction vector pointing from the robot towards the expected position of $\mathbf{q}_{test}$.

Let further $\mathbf{v}^{GT}_{test} = \left(\mathbf{p}^{GT}_{test} - \mathbf{r}\right) / ||\mathbf{p}^{GT}_{test} - \mathbf{r}||$ be the normalized direction vector pointing from the robot's position towards the known, true position of $\mathbf{q}_{test}$. With that, the *azimuth error* in each scenario i becomes

$$Err^{\phi}_i = \left|\arccos\left(\mathbf{v}^{GT}_{test} \cdot \mathbf{v}^E_{test}\right)\right| \, . \tag{2.14}$$

For good localization performance, Err^{ϕ}_i should be as small as possible for all test scenarios. Figure 2.5 shows a histogram of the azimuth errors acquired in the randomly generated test scenarios described above. Obviously, localization is constantly precise, with a mean error of 0.0103 rad (0.5889 °) and a standard deviation of 0.0192 rad (1.0985 °). Note that the outliers in Fig. 2.5 can likely be attributed to the challenging quality of the employed stimuli.

2.3 Active exploration – building environmental maps

(The following relates to c1, c4, and c6)

Besides the formation of audio-visual objects and multi-modal feedback, the Two!Ears project aims at a system architecture that allows for *active exploration.* With this goal in mind, the robot has to wander the environment in order to acquire additional information about the scenario. In the current context, active exploration will be employed to retrieve the world coordinates of each active sound source in the x–y plane.

Assume, as above, that $\mathcal{Q} = [\mathbf{q}_1, ..., \mathbf{q}_{N_Q}]$ is the set of all active (static) sources in the LVTE. Further, let $\mathbf{p}_i^{GT}$ define the ground-truth position of acoustic source, i, in the azimuthal plane. Note that currently a bijective relation between the physical sound sources and the emitted acoustic stimuli is postulated. That is, if any $\mathbf{q}_i$ emits a stimulus of category K_i^A, stimuli emitted by all other sound sources have to be chosen from $\mathbf{S}_V \setminus K_i^A$. In conclusion, define $\mathbf{r}_t$ as the position of the robot at time t.

All sources in $\mathcal{Q}$ are activated in a sequential, repetitive pattern of the following form, $\mathbf{q}_1 \rightarrow \mathbf{q}_2 \rightarrow \ \dots \mathbf{q}_{N_Q} \rightarrow \mathbf{q}_1 \rightarrow \ \dots$. Intervals between consecutive source activations are set to 0.2 s. This pattern is required to let *ReactToStimulusKS* cleanly lock on an active source, thereby providing reliable category/azimuth information. Enhanced localization/segmentation methods – such as sketched in D3.2 – will be tested in later system versions in order to allow for simultaneous activation of multiple sources.

2.3.1 Knowledge sources – definition and binding pattern for environmental map formation

Applying the LVTE/blackboard system to the active-exploration domain requires a number of additional knowledge sources, which are assessed in detail below. Adding up to that, a redesign of the blackboard wiring – shown in Fig. 2.4 – becomes necessary. The required modifications are also described in the following discussion.

MoveToKS

MoveToKS inherits from *AbstractKS* and reads from the blackboard-memory *positionRequest* entries, generated by *MemoryFormationKS* described below. Assume that a given position request demands the robot to reach some goal position, $\mathbf{p}^{goal}$. Then *MoveToKS* triggers robot motion in the LVTE framework, guiding the robot towards $\mathbf{p}^{goal}$ along a a straight line. Mind that in the current version the robot follows this path as if it were equipped with *omniwheels* in order to significantly facilitate motion planning. In later system versions, more complex motor patterns will be installed.

Similar to rotation, translatory motion is executed in a completely transparent manner, that is, the blackboard is not blocked while the robot platform approaches $\mathbf{p}^{goal}$. The system always computes the shortest translatory path, taking into account the maximum linear velocity of the robot device. Once $\mathbf{p}^{goal}$ is reached, the LVTE broadcasts a notification event that is received by *MemoryFormationKS*.

MemoryFormationKS

Descending from *AbstractKS*, *MemoryFormationKS* generates memory representations encoding the auditory categories and positions of the sound sources that are active in a given scenario. Assume that the knowledge source maintains a *memory set*, $\mathcal{M}$, which expands with the arrival of novel *memory patterns*. To actually form such memory patterns, results from *ReactToStimulus* knowledge source are monitored. Given an auditory object hypothesis, $\mathcal{H}^{o}_{i,h} = \{\phi_{i,h}, K^{A}_{i,h}\}$, observed at time t_h for some putative source, $\mathbf{q}_i$, the short-term memory compares $K^{A}_{i,h}$ to the categories of all memory patterns already available in $\mathcal{M}$. If the category is not found, a new memory representation, $\mathbf{m}_{K^A_i}$, is formed and primed with the *observation* $\mathbf{o}_h = \{\mathbf{r}_h, \phi_{i,h}\}$ – where $\mathbf{r}_h \equiv \mathbf{r}_{t_h}$ for notational simplicity. Eventually, $\mathbf{m}_{K^A_i} = \{\mathbf{o}_h\}$ is appended to $\mathcal{M}$. For notational convenience assume that $\mathbf{o}^j$ represents the j^{th} element of the memory set and $K(\mathbf{o}^j)$ retrieves the auditory category that $\mathbf{o}^j$ belongs to.

If, however, the category $K^{A}_{i,h}$ can be retrieved from the memory-data set, all observations $\mathbf{o}^j = \{\mathbf{r}^j, \phi^j\}$ in the corresponding entry, $\mathbf{m}_{K^A_i}$, are compared to the new observation $\{\mathbf{r}_h, \phi_{i,h}\}$, that is,

$$\arg\max_j \left|\mathbf{r}_h - \mathbf{r}^j\right| < 0.5 \quad \wedge \quad \arg\max_j \left(\arccos\left(\mathbf{v}^h \cdot \mathbf{v}^j\right)\right) < 0.0175\,, \tag{2.15}$$

with $\mathbf{v}^h = [\cos(\phi_{i,h}), \sin(\phi_{i,h})]^T$ and $\mathbf{v}^j = \left[\cos(\phi^j), \sin(\phi^j)\right]^T$.

If Eq. 2.15 holds, $\mathbf{o}_h$ comprises no new*information content* and can safely be discarded. Otherwise, the corresponding memory representation has to be augmented according to $\mathbf{m}_{K^A_i} = \mathbf{m}_{K^A_i} \cup \{\mathbf{r}_h, \phi_{i,h}\}$. This basic form of *non-trival learning* [29] keeps the memory footprint as small as possible while simultaneously ensuring correct memorization of all overheard category-related information.

Note that in the current version, release of previously acquired information (*forgetting*) is not implemented in *MemoryFormationKS*. This feature is envisaged for forthcoming system versions together with the evolution of a *ShortTermMemoryKS* and a *LongTermMemoryKS*.

The data stored in $\mathcal{M}$ can be used for *source triangulation.* Assume that sound source $\mathbf{p}_i$ emits a stimulus of category K_i^A. Postulating that $\mathbf{m}_{K_i^A}$ comprises more than one observation and recalling the bijective correspondence between auditory cateories and physical sources, the most likely position of the sound source corresponding to $\mathbf{m}_{K_i^A}$ can be inferred in the x–y plane. To that end, information from all $\mathbf{o}^j = \{\mathbf{r}^j, \phi^j\} \in \mathbf{m}_{K_i^A}$ is used. Let $\mathbf{l}^j = \mathbf{r}^j + l\left[\cos(\phi^j), \sin(\phi^j)\right]^T$ represent a line emanating from the robot's position in observation j and pointing along the direction vector derived from ϕ^j. Finding the approximate intersection point of all $\mathbf{l}^j$ – by means of least-square estimation – then yields the most probable origin, $\mathbf{p}_i$, of the stimulus related to auditory category K_i^A. As the auditory categories are bijectively coupled to the sound sources – compare above – $\mathbf{p}_i$ is also the position of sound source $\mathbf{q}_i$. The list of inferred positions for all overheard sources is eventually pushed to the blackboard memory as a *triangulatedLocations* entry.

Note that the triangulation scheme as decribed above will fail without active exploration. Given a passive robot and static sources, the KS would still learn all overheard auditory categories. However, no $\mathbf{o}^j$ with $j > 1$ would pass the criterion in Eq. 2.15, thus negating the triangulation precondition $\left|\mathbf{m}_{K_i^A}\right| > 1$.

Consequently, *MemoryFormationKS* triggers an exploratory motion of the robot if, (a), no further observation was made for $T_D = 5\,\mathrm{s}$ or, (b), observations for all active sources were processed with regard to the current robot position. To enable evaluation of the latter, N_q is passed to the knowledge source and the bijective correspondence between sources and stimuli is exploited.

Computation of the optimal direction for the triangulation path follows an information-maximization paradigm as follows. Given that the above condition (b) holds for the first time, the putative azimuth angles for all observed sources are approximately known and can be retrieved from $\mathcal{M}$. To that end, let ϕ_k^1 describe the azimuth extracted from the first observation in $\mathbf{m}_{K_k^A}$. With that, define a set of vectors, $\mathbf{v}_k = \left[\cos(\phi_k^1), \sin(\phi_k^1)\right]^T$ with $k = 1 \ldots N_Q$. The unit vector of the optimal triangulation path is then chosen according to

$$\mathbf{v}^{opt} = \arg\min_{\mathbf{v}} \sum_{k=1}^{N_Q} \left(\mathbf{v}_k \cdot \mathbf{v}\right) . \tag{2.16}$$

This practice generates a triangulation path that is 'as perpendicular as possible' to all given $\mathbf{v}_k$ and thus maximizes the efficiency of follow-up triangulation. The length of the path is currently chosen as $\left\|\mathbf{v}^{opt}\right\| = 1.0\,\mathrm{m}$. For completeness note that com-

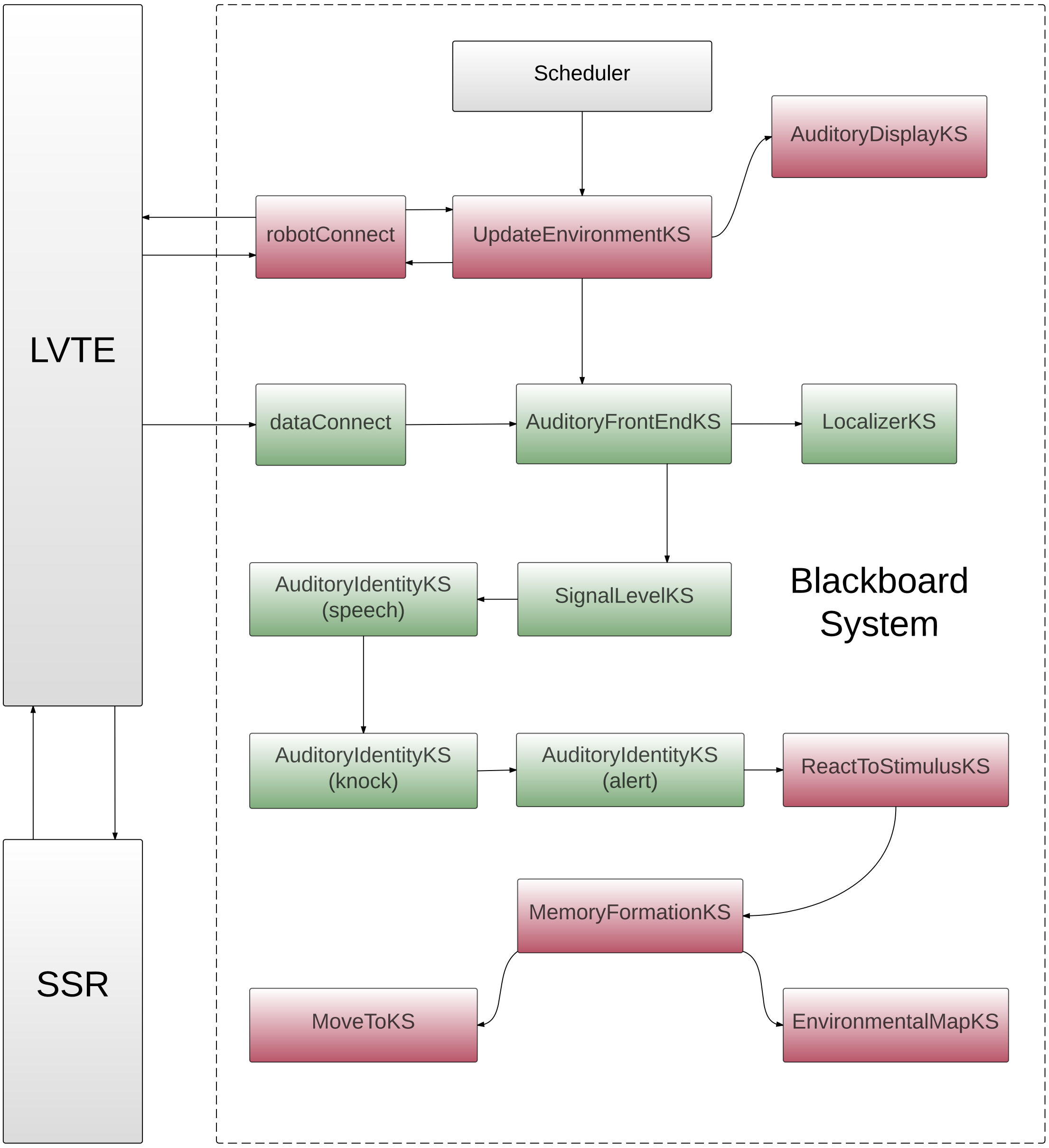


Figure 2.6: The basic LVTE/blackboard system configured for solving the multi-source triangulation task (see running text). Colors indicate additional connectivity. Red means that the tinted KS can communicate with LVTE via the robotConnect variable of the blackboard. Green means that the corresponding KS is dependent of the AFE connection of the blackbord

putation of ideal exploration paths will become even more reliable if $\left|\mathbf{m}_{K_i^A}\right| > 2$, with $i = 1, ..., N_Q$.

The computed path is then transformed into a *positionRequest* and pushed onto the blackboard. At this point, *MoveToKS* takes over and guides the robot to the goal position determined by $\mathbf{r}_{cur} + \mathbf{v}^{opt}$, where $\mathbf{r}_{cur}$ is the current position of the robot.

It may be argued that *MemoryFormationKS* should not be responsible for the computation of ideal exploration vectors. Therefore, future system versions will comprise a dedicated *ActiveExplorationKS* that autonomously plans exploration paths using data from *MemoryFormationKS*.

EnvironmentalMapKS

To visualize the results achieved by active exploration, *EnvironmentalMapKS* monitors the blackboard memory for *triangulatedLocations* provided by *MemoryFormationKS*.

Once that such an entry has been found, a bivariate Gaussian, $\sigma^2 = 1.0$, is placed at each retrieved location in a two-dimensional *probability map representation* of the environment. Note that, currently, the measured locations are not weighted according to their reliability. Fig. 2.7 shows results from environmental-map formation in a basic triangulation scenario.

Currently, *EnvironmentalMapKS* acts as a plain visualization unit. This role is intended to be changed in upcoming system versions by augmenting the environmental-map knowledge source with capabilities regarding multi-modal-cue fusion, autonomous map analysis and information weighting.

As in audio-visual-object formation, it remains to provide the wiring scheme used to interconnect the above knowledge sources. Figure 2.6 provides an overview of the binding structure employed for triangulation and environmental map formation. Again, arrows indicate the firing sequence for the single knowledge sources. Note that, as above, the depicted blackboard scheduling is experimental and will be re-assessed in future system versions.

2.3.2 Experiments regarding multi-source triangulation

To evaluate the precision of the proposed multi-source localization method, $N_T = 100$ simple test scenarios (anechoic conditions, 10 x 10 m) are generated. Assume that $N_Q = 3$ sources, $\mathbf{q}_{1...3}$, are randomly distributed in every created scene. Let $\mathbf{p}^{GT}_{1,...,3}$ represent the ground-truth coordinates of $\mathbf{q}_{1...3}$ in the x–y plane. Note that these coordinates, despite

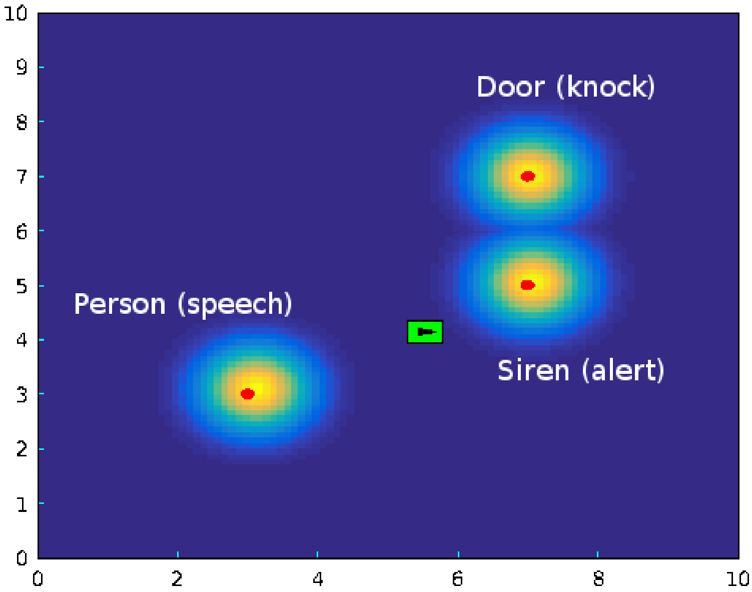


Figure 2.7: An environmental map computed by *EnvironmentalMapKS*, using data from *MemoryFormationKS*. The image shows estimates for a typical triangulation scenario. For visual clarity, the true positions of all observed sources are overlaid (red dots) as well as the robot's position (green rectangle). Source labels and auditory categories $\{$**'speech'**, **'alert'**, **'knock'**$\}$ are manually provided for completeness. Estimated sound-source positions are encoded as a 'heat-map', whereby blue tones indicate low probability for the presence of a source and yellow tones represent likely source locations. Room dimensions are 10 x 10 m

their random nature, are forced to stay within a concentric ring with an inner radius of $r_i = 2\,\mathrm{m}$ and an outer radius of $r_o = 4\,\mathrm{m}$. The minimum azimuth difference between two neighboring sources is Δ_ϕ degree. The center of the ring is identical to the robot's initial position, $\mathbf{r} = [5, 5]^T$.

This practice allows the robot to move freely within the perimeter described by r_i and prevents sources to be placed too close to the periphery of the scene. Setting Δ_ϕ to 30 ° ensures that the localizer will be able to cleanly lock on to each perceived sound source. The activation schedule of the sources is sequential, using $\mathbf{q}_1 \rightarrow \mathbf{q}_2 \rightarrow ...\mathbf{q}_{N_Q} \rightarrow \mathbf{q}_1 \rightarrow \ldots$ as its standard pattern.

Intervals between source activations are chosen to equal 0.2 s, allowing *ReactToStimulusKS*, and *MemoryFormationKS* to reliably process each overheard stimulus. With that, let $\mathbf{p}^E_{1...3}$ represent the positions of the sound sources as estimated by the triangulation system. The

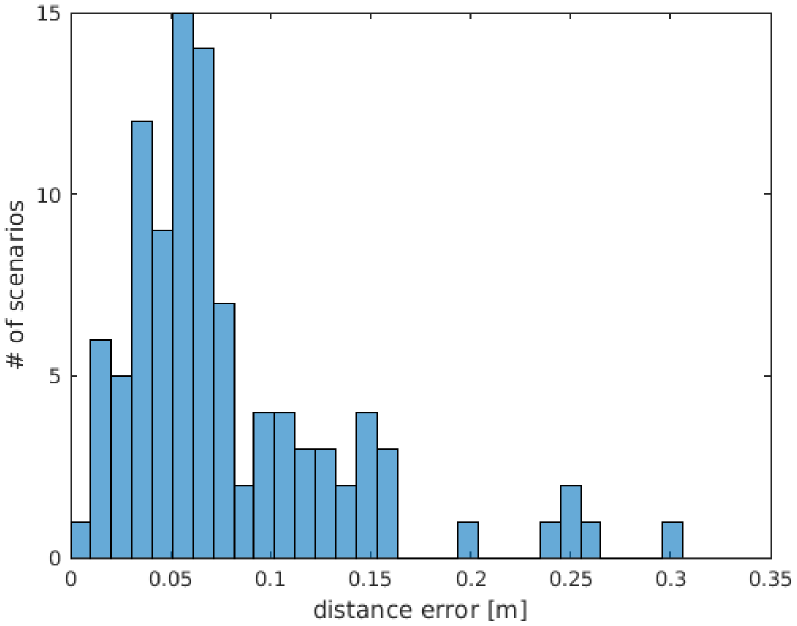


Figure 2.8: Experimental results from triangulation experiments. The distance error accumulates around 0.0793 m. See running text for details

position estimation error for scenario i is then described by

$$Err_i^{Pos} = \frac{1}{N_Q} \sum_{j=1}^{N_Q} ||\mathbf{p}_j^E - \mathbf{p}_j^{GT}|| \,. \tag{2.17}$$

Figure 2.8 shows the distribution of Err_i^{Pos} for all N_T scenarios.

The position estimation errors accumulate about 0.0793 m, with a standard deviation of 0.0578 m. This documents a fair precision of the proposed multi-source triangulation scheme. Note that the error metric in Eq. 2.17 does not disclose outliers with respect to single sources. Indeed, such rare outliers exist. However, they are attributed to weak stimulus quality and/or geometric source configurations that cannot reliably be triangulated with the one-step exploration pattern proposed above. In upcoming system versions, multi-step exploration will alleviate this issue.

2.4 Attention-controlled head turning

(The following relates to c1, c4, and c6)

2.4.1 Introduction

A low-level attention module, *HeadTurningModulationKS* (HTMKS), was developed that aims at modulating the head movements of the robot by inhibiting purely reflexive behavior, such as moving the head every time that a sound pops up. Inhibition of this reflexive movement relies on the following principle. *The more an object is observed in an environment, the more likely it is to appear in the future.* Thus, HTMKS represents a real-time learning algorithm that makes the robot learn the probability distribution of all the objects that have been observed during the exploration of an environment. This learning enables the robot to steer its attention (by means of head turning) to important sound sources only, that is, to sound sources that have a low probability of occurrence. This notion of importance was formalized through the concept of *Congruence* of an object with regards to the explored environment – compare Sec. 2.4.4. In addition, *HeadTurningModulationKS* embeds a multi-modal-fusion algorithm that can correct wrong audio-visual inputs or infer a missing modality. Further, the algorithm triggers head movements in order to achieve a step-by-step refinement of modality inference.

The *HeadTurningModulationKS* descends from the *AbstractKS*, and consists of two modules: the *Multi-modal Fusion and Inference module* (MFImod), and the *Dynamic Weighting module* (DWmod) – as described in [9, 43]. The focus is computed separately by these two modules and the results are then fused in order to trigger a head movement. Figure 2.9 shows the integration of the HTMKS within the blackboard system, together with a simplified scheme of the internal architecture of the HTMKS.

2.4.2 Definitions

Before describing these two modules, it is necessary to introduce the definitions and notations that they rely on. Let $\mathcal{R}$ and $\mathcal{E}$, respectively, be the robot and environment sets with

$$\mathcal{E} = \{e^{(1)}, e^{(2)}, \ldots, e^{(N_e)}\}\,, \tag{2.18}$$

where $e^{(i)} \in \mathcal{E}$ represents the i^{th} environment explored by $\mathcal{R}$, and N_e represents the number of considered environments. Each environment, $e^{(i)}$, is defined as a set of objects, o_j,

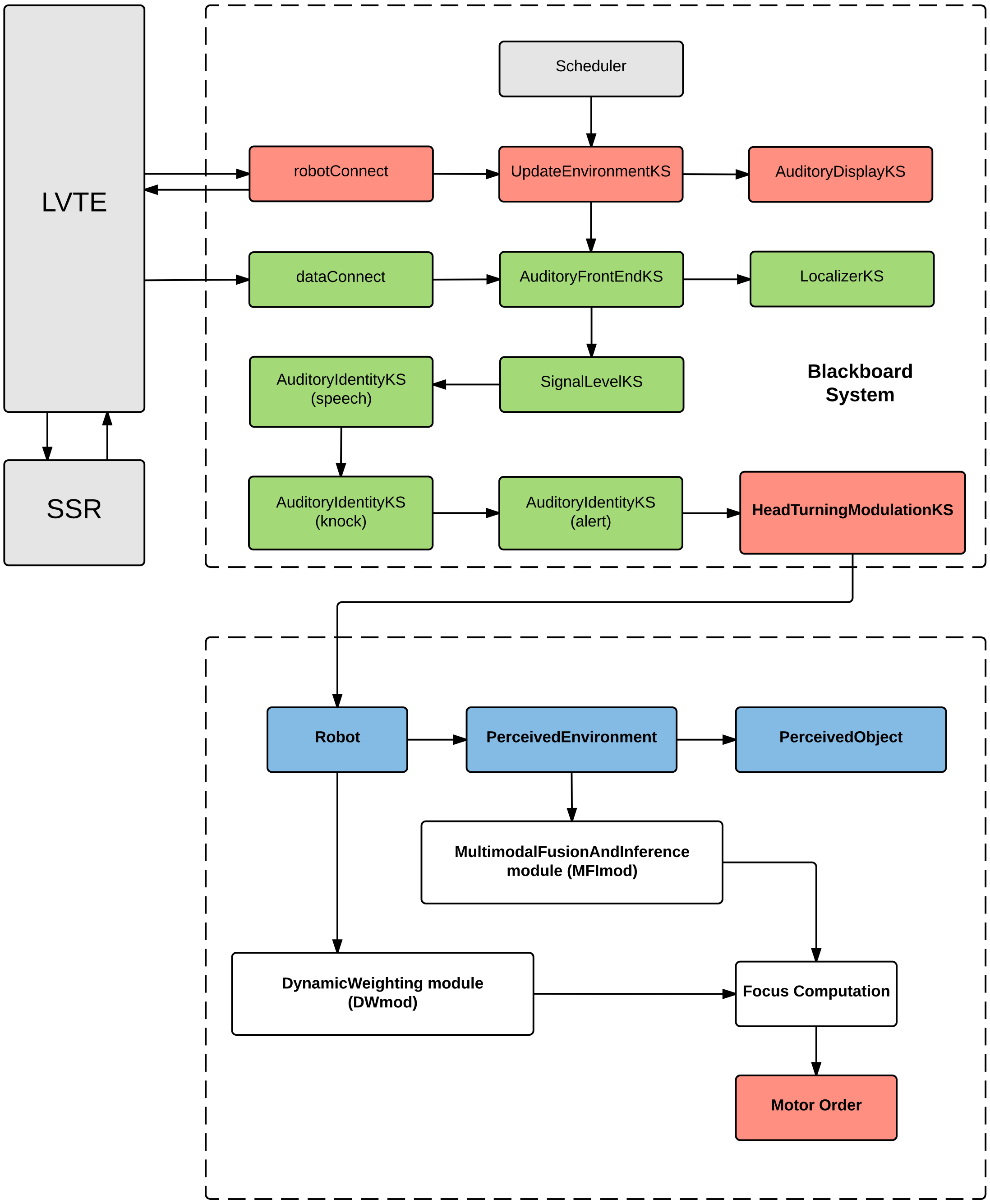


Figure 2.9: The basic LVTE/blackboard system configured for testing the *HeadTurningModulationKS*. Red and green colors indicate additional connectivity: red means that the tinted KS can communicate with the LVTE, via the robotConnect variable of the blackboard. Green means that the corresponding KS is dependent on the AFE connection to the blackboard . Blue color indicates MATLAB®classes

such that

$$e^{(i)} = \{o_1, o_2, \dots, o_{N_i}\}, \tag{2.19}$$

with N_i the number of detected objects in the environment, $e^{(i)}$. Each object, o_j, is defined by its relative angle to the robot, θ_j, an auditory label, a_j, and a visual label, v_j, so that

$$o_j = \{\theta_j, a_j, v_j\}. \tag{2.20}$$

The relative angle, θ_j, between the object and the robot is provided by *LocalizerKS*. The multi-modal labels, a_j and v_j, are provided by *AuditoryIdentityKS* and *VisualIdentityKS* and will be sent to MFImod. Now define the audio-visual categories $c^{(i)}(a,v)$ of the i^{th} environment by

$$c^{(i)}(a,v) = \{o_j \in e^{(i)}, a_j = a, v_j = v\}. \tag{2.21}$$

The term $c^{(i)}(a,v)$ basically represents the collection of objects sharing the same auditory and visual labels a and v respectively. All categories of the i^{th} environment are gathered into a set of categories $\mathcal{C}^{(i)}$ such that $\mathcal{C}^{(i)} = \{c^{(i)}(a,v)\}$.

2.4.3 Multi-modal fusion and inference module

The multi-modal fusion and inference module (MFImod) is responsible for the creation of relevant audio-visual objects as needed by the *Dynamic Weighting module* – see Sec. 2.4.4. It receives data from both *AuditoryIdentityKS* and *VisualIdentityKS*. MFImod is a real-time-learning algorithm that aims at associating audio stimuli with visual stimuli and *vice versa* by addressing the two following questions.

- How can bad or wrong classifier output be dealt with?
- How can the robot internally create a multi-modal object when one modality is missing?

Actually, before transmitting outputs from classification experts to DWmod, these data have to be correct. If not, the reaction of the robot will be erroneous. MFImod consists on two distinct artificial neural networks, (i), a *self-organizing map* (SOMnet) and, (ii), a *Multilayer Perceptron* (MLPnet). MFImod can be described as an *autosupervised* learning algorithm. SOMnet will create audio-visual categories from output of the classification experts. MLPnet will then use these categories to learn the coupling between audio and visual data. No knowledge is thus put into the robot before it experiences an environment.

At time step $t = n$, MFImod receives two vectors of probabilities, one from *AuditoryIdentityKS*, denoted $a[n]$, and one from *VisualIdentityKS*, denoted $v[n]$. Then a vector $av[n]$ is defined as the vector resulting from concatenation of $a[n]$ and $v[n]$.

If $av[n]$ contains both audio and visual information, it is added to the matrix AV that gathers all the previous audio-visual inputs received. The whole matrix, AV, is then sent to SOMnet[1], which will cluster it in order to create audio-visual categories, $c^{(i)}(a, v)$, without any prior knowledge. Once these audio-visual categories have been created, MLPnet is trained with the same input vector, $av[n]$, but with a slight modification, namely, an *information-masking* step is applied to it. This is computed by creating two new input vectors from the audio-visual one, one with the audio information removed and the other one with the visual information removed. These removals are achieved by setting all the corresponding components to zero. To train MLPnet, the categories $c^{(i)}(a, v)$ created by the SOMnet are used as the targets – this is why the system is defined as *auto-supervised*. The training phase corresponds to learning of the association of auditory and visual information.

Consequently, at every time step, the following two cases are possible.

- Audio and visual information **is available**. Then the input vector, $av[n]$, is used to train both the SOMnet and the MLPnet
- Audio or visual information **is missing**. Then the input vector, $av[n]$, is sent directly to the MLPnet to infer the audio-visual category

Note that if audio-visual information is available but the classification experts exhibit low probabilities, what means untrustful classification, the input vector, $av[n]$, will anyway be sent to the MLPnet to be corrected.

To sum up, one step of learning consists of the following two components.

1. Creation of audio-visual categories as are derived from the data observed during active exploration
2. Learning of associations between modalities

After sufficient learning, the module is able to, (i), correct information from the classification experts, if they are not trustful enough and, (ii), infer a missing modality. At each time step, the audio-visual category created, $c^{(i)}(a[n], v[n])$, will be fed into the *Dynamic-Weighting module* (DWmod) – see Sec. 2.4.4.

The *Multi-modal-Fusion-and-Inference* module has one major advantage and one major drawback. The most striking advantage is that it is unsupervised and highly flexible, since it creates knowledge for the robot that does not depend on prior information. The main drawback is, however, that it is prominently dependent on initial exploration of the

[1] Note that, for the moment, no memory has been implemented, since the duration of the simulations is quite short – about some minutes with around 30 objects present in the environment

environment.

2.4.4 Dynamic-weighting module

This module receives the inferred audio-visual category from the *Multi-modal-Fusion-and-Inference* module. DWmod is responsible for the modulation of the head-turning reflex triggered by auditory stimuli. The aim of DWmod is to prevent the robot from making ceaseless head movements towards any new sound source that occurs in the environment. This is achieved by weighting in real time all the perceptual objects detected by the robot, according to their importance. Importance is here formalized by the notion of *Congruence* defined along

(a) Features shared by two perceptual objects, such as visual and auditory labels

(b) Links that exist between a perceptual event and a given environment

If an object has been detected as incongruent, a quick head movement will be triggered towards the direction of the object. This head movement will have several consequences, which together lead to more accurate perception of the object. These consequences are

(i) Enhancing of estimated position of the object by updating its ITDs and ILDs

(ii) Enhancing the discrimination of the object from other sound sources present in its surroundings – in other words, address the problem of sound-source separation as observed in the *Cocktail-Party Problem*

(iii) Accessing missing visual information regarding the object

In order to decide if an object, o_j, is of interest, each object a weighting function, $w(o_j)$, is associated to. In all of the following, an audio-visual object, o_j, will be classified as incongruent *if other objects belonging to the same category, $c^{(i)}(a_j, v_j)$, have not been detected by the system in the past.* The classification of congruent versus incongruent objects is based on the object-weighting function, $w(o_j)$, with $w(o_j) \in [-1; 1]$. Hereby $w(o_j) = -1$ represents a highly **congruent** object, while $w(o_j) = 1$ indicates a highly **incongruent** object. Note that the former case will not trigger any movement of the robot while the latter will.

First, and based on the previous definitions, lets define the pseudo-probability, that is, the statistical frequency, $p(c^{(i)}(a_j, v_j))$, as follows,

$$p\left(c^{(i)}(a_j, v_j)\right) = \frac{|c^{(i)}(aj, v_j)|}{N_i}\,, \tag{2.22}$$

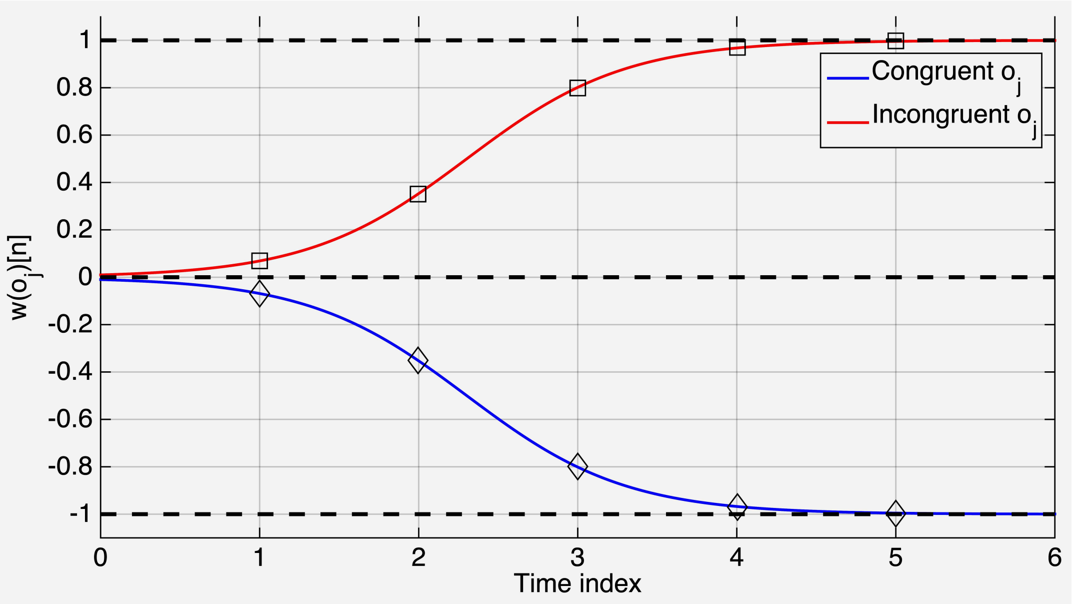


Figure 2.10: Object weight, $w(o_j)[n]$, as a function of time. Depending on the congruence of the object, one of the two functions is selected. Dots indicates the discrete time steps where values are selected

with

$$\sum_{n=1}^{|\mathcal{C}^{(i)}|} p(c^{(i)}(a_n, v_n)) = 1\,, \tag{2.23}$$

where $|.|$ denotes the set cardinality. The pseudo-probability, $p(c^{(i)}(a_j, v_j))$, denotes the likeliness of the occurrence of an object, o_j, that belongs to category $c^{(i)}(a_j, v_j)$. On this basis, the weight, $w(o_j)$ of the object o_j is defined by two smooth symmetric sigmoid functions lying in the range of $[-1; 1]$, with

$$w(o_j)[n] = \begin{cases} 1/(1+100\,e^{-2n}) & \text{if } p(c^{(i)}(a_j, v_j)) < K, \\ 1/(1+0.01\,e^{2n}) - 1 & \text{else,} \end{cases} \tag{2.24}$$

where K_i denotes a frequency threshold and n represents the time-frame index. $w(o_j)[n]$ is plotted in Fig. 2.10 as a function of time index. Eq. (2.24) clearly shows the relation between a high object weight, $w(o_j)$, and a low probability of occurrence of the object's category, $p(c^{(i)}(a_j, v_j))$. Thus, if object o_j appears in the current scene, it will be categorized as *incongruent*, and a motor command will be triggered. The threshold, K, is set to

$$K_i = \frac{1}{|\mathcal{C}^{(i)}|}, \tag{2.25}$$

that is, to $w(o_j) = 1$ (which denotes incongruency), if the probability, $p(c^{(i)}(a_j, v_j))$, is smaller than a random choice among equiprobable categories.

For a frame length of $T_w = 20\,\mathrm{ms}$, one then gets $w(o_j)[n] \approx 1$ at time $t = 100\,\mathrm{ms}$. As a last step, once the weight, $w(.)$, of a new object has been computed, it has to be decided whether a motor command has to be triggered. A motor command, $m[n]$, is produced to let the robot's head turn to the current sound source at time index n, according to

$$m[n] = \begin{cases} 1 & \text{if } w(o_j)[n] > 0.98, \\ 0 & \text{else}. \end{cases} \tag{2.26}$$

A threshold value of 0.98 was selected due to the value of the weighting function, $w(.)$, at 100 ms – see Eq. 2.24.

2.4.5 Simulation and results

Simulations were conducted in framework of LVTE. The results of focus computation using *solely* DWmod have been presented in [9]. In the experiment reported here, DWmod was used in couple with MFImod, thus, only the resulting output of this combination is discussed here.

A free field environment populated with three sources was set up in order to test HTMKS – see Fig. 2.11. Every source had a dedicated visual label, but audio stimuli were randomly emitted by one of the three sources. Three audio stimuli ('*speech*, '*knock*', and '*alert*') and three visual stimuli ('*person*', '*door*', and '*siren*') were chosen to test the HTMKS for a 60-s-long simulation. 24 audio-visual objects were randomly created with a rate of correct audio-visual pairs of 80 %. The correct AV pairs are the following: *person/speech*, *door/knock*, and *siren/alert*. Every other AV pair is thus considered to be erroneous.

Figure 2.12 shows the results of a single simulation. In this scenario, the HTMKS (blue solid line) was tested versus a naïve robot that would turn its head towards every new sound source (black dashed line). The first result was the number of head movements triggered, that is, 16 for the HTMKS versus 24 for a naïve robot. It can be seen that the first inhibition of a reflexive movement occured early, namely, at the 14^{th} object – from $t = 33.25\,\mathrm{s}$ to $t = 34.69\,\mathrm{s}$. The red line shows that MFImod did not need further information about this object at this time. Consequently, the focus computation was only based on DWmod. However, since the object was detected as being congruent with respect to what the robot already had observed, no head movement was triggered.

The second interesting result is about the correction of wrong audio-visual pairs. As to this simulation, four objects out of 24 were created with a wrong combination of audio

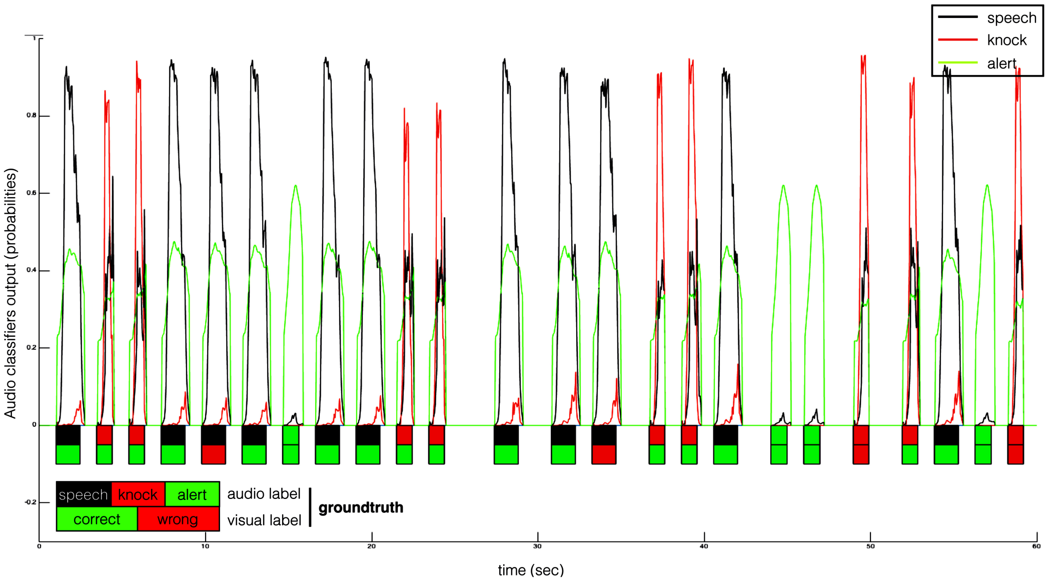


Figure 2.11: Output of the audio classifiers during the 60 -s simulation. The boxes indicate the onset and offset time of the objects populating the environment. The upper color denotes the ground-truth knowledge – see legend. The lower color indicates whether the labels of the objects have been picked up from among the set of correct AV pairs (green color) or from among the erroneous set (red color)

and visual pairs, nameley, ≈ 16.6 %. The erroneous objects are # 5, 14, 20 and 24. In Figure 2.12, the black asterisks indicates that HTMKS – via MFImod computation – was able to correct these wrong AV pairs. Further, for objects # 14, 20 and 24, MFImod was able to infer the missing label without turning the head. This means that MFImod did not allow the robot to get the wrong visual information. Thus, in order to verify the robustness of the MFImod when faced with wrong AV pairs, the performance of inference with erroneous input vectors was tested – for instance, *door/speech*.

As an example, the case of object # 14 was assessed by using the mean data of this object, that is, $[0.6064, 0.0145, 0.3790, 0, 0, 0]$. The first three components correspond to respective audio-classifier outputs, namely, from the *speech*, *knock*, and *alert* classes. The last three components correspond to the visual-classifier output, that is, the *person*, *door*, and *siren* classes. The latter are equal to zero since no head movement was triggered towards the direction of this object. These three components were then replaced with the set of values consigned in Tab. 2.1. The two possibilities for wrong-label combinations were tested, namely, *door/speech* and *siren/speech*. Both were presented with two different percentages of confidence – 60 % and 80 %.

The results presented in Tab. 2.1 show that even with wrong audio-visual pairs, the system

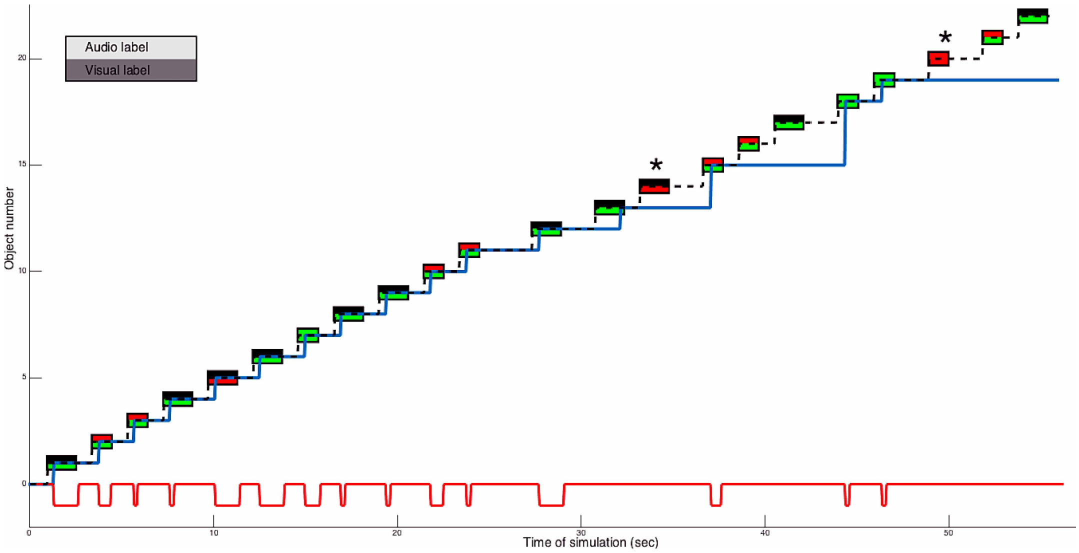


Figure 2.12: Results of the focus computation by the *HeadTurningModulationKS*. Blue line: objects the robot is facing. Black line: behavior of a purely reflexive robot. Red line: module responsible for the focus computation (0: DWmod, -1: MFImod). The boxes indicate the objects and their labels. Asterisks indicates a good correction of a wrong AV pair of labels by the HTMKS (through the MFImod).

Input vector		**Output vector**	**Inferred AV class**
audio components	visual components		
$[0.606, 0.014, 0.379]$	$[0.2, 0.2, \mathbf{0.6}]$	$[0.14, 0.19, \mathbf{0.65}, 0.00, 0.00, 0.00]$	*person/speech*
	$[0.1, 0.1, \mathbf{0.8}]$	$[0.25, \mathbf{0.74}, 0.00, 0.00, 0.00, 0.00]$	*door/knock*
	$[0.2, \mathbf{0.6}, 0.2]$	$[0.00, 0.00, \mathbf{0.99}, 0.00, 0.00, 0.00]$	*person/speech*
	$[0.1, \mathbf{0.8}, 0.1]$	$[0.00, 0.00, \mathbf{0.99}, 0.00, 0.00, 0.00]$	*person/speech*

Table 2.1: Inference of the audio-visual label by MFImod with simulated visual components for object # 14. The values in bold indicates the highest component of the vector – values are rounded to the 2^{nd} decimal. The first three components correspond to respective audio-classifier outputs, that is, *speech*, *knock*, and *alert*. The last three components denote the visual-classifier outputs, that is, *person*, *door* and *siren*

was able to perform a correction in a relevant way. For object #14, the audio label was '*speech*'. It can be seen that for the input vectors # 1, 3 and 4, the system output corresponded to the audio-visual category '*person/speech*'. This holds in particular for vector # 4, in which the '*door*' component was set to a very high value. However, for vector # 2, the system corrected the input by inferring the '*door/knock*' label. This result shows that the system works fine. Indeed, it inferred an audio-visual category that belongs to the correct-pair set as defined at the beginning of the simulation. The reason for the inference not being correct was that the visual component had taken lead over the audio component.

2.4.6 Discussion

Results from the first simulations are very encouraging and constitute a first proof-of-concept for *HeadTurningModulationKS*. A knowledge source is thus available that is able to inhibit reflexive movements of a naïve robot in order to let it steer its attention to important objects. In addition, HTMKS embeds a multi-modal-fusion module that can correct wrong audio-visual data or infer a missing modality. The main advantages of this system are, (i), that it continuously learns and, (ii), that it is multi-modal (Actually, it is conceptually possible to add any source of information into MFImod) and, (iii), that no prior knowledge has to be put into the system, thus making it highly flexible and more relevant in respect of the bio-inspired paradigm that Two!Ears adopts.

Ongoing work on the Head-Turning-Modulation approach as implemented on HTMKS is focused on testing it in more complex and realistic scenarios, particularly in the following.

- More types of sound and visual stimuli
- Overlapping events
- New environments to learn
- Additional information taken into account, such as position, loudness and speed
- Longer simulations
- Short-, mid- or long-term-memory implementation

2.5 Formation of attention and attention-based control of feedback processes

(The following relates to b1, c1)

2.5.1 Background

Human listeners must answer two questions in order to fully understand an acoustic scene, namely, *what* the sound sources are, and *where* they are. In machine hearing, these two issues have been addressed by many studies via computational approaches for sound-source separation, classification and localisation [44]. However, machine systems for answering 'what' and 'where' questions are typically much less tightly-integrated than they appear to be in biological hearing. Work in Two!Ears has addressed this issue by developing an approach for binaural localisation that exploits top-down knowledge about the spectral characteristics of sources in acoustic scenes.

A number of psychophysical studies have found evidence for top-down effects in sound localisation. For example, covert shifts of attention can reduce reaction times when the spatial location of a target sound is cued by a preceding sound [36]. Physiological studies have also shown that sound localisation can be modulated by top-down influences. In the barn owl, sound localisation (including orienting behaviour such as head and body movements) is influenced by selective attention at the level of the midbrain; neural responses associated with the location of behaviourally relevant stimuli (such as a food source) are enhanced [15]. Similarly, neural circuitry for gaze control exerts a top-down influence on the responsiveness of auditory neurons that are tuned to specific spatial locations [45]. Taken together, these findings suggest the existence of cross-modal mechanisms for top-down gain control of spatial hearing.

2.5.2 System description

A framework for sound localisation is proposed in which information from source models is used to selectively weight binaural cues. The system therefore combines top-down and bottom-up information flow within a single computational framework. By exploiting source models in this way, sound-localisation performance can be improved under conditions in which multiple sources and room reverberation are present.

The auditory front-end (AFE) of Two!Ears was employed to analyse binaural ear signals, consisting of a bank of 32 overlapping Gammatone filters with centre frequencies uniformly spaced on the ERB scale between 80 Hz and 8 kHz [44]. Inner-hair-cell function was approximated by half-wave rectification. Afterwards, the cross-correlation between the right and left ears was computed independently for each frequency channel using overlapping frames of 20 ms duration with a shift of 10 ms.

Two primary binaural cues, ITD and ILD, were extracted as features for binaural localisation. The ITDs were estimated as the lag corresponding to the maximum in the cross-correlation function output. The ILD corresponded to the energy ratio between the left and right ears within the analysis window – expressed in dB. The pair of ITD/ILD features was estimated for each frequency channel independently, to form a 2–D-localisation feature vector, $\boldsymbol{o}_{tf}$, for time frame t and frequency channel f.

Source spectral characteristics were modelled using ratemap features [6]. A ratemap is a spectro-temporal representation of the auditory nerve-firing rate, extracted from the inner-hair-cell output of each frequency channel by leaky integration and downsampling. For the binaural signals used here, the ratemap features were computed for each ear and then averaged across the two ears. They were finally log-compressed to form 32 feature vectors, $\boldsymbol{x}_t$. All these processing steps were executed within the Two!Ears software system.

Gaussian mixture models (GMMs) were then used to model the relations between the binaural features and corresponding azimuth angles statistically. 72 azimuth angles, ϕ, in the full 360 ° azimuth range (5 ° steps) were considered. A separate set of GMMs, λ_f^ϕ, was used for each frequency channel, f. Given the observed localisation-feature vector, $\boldsymbol{o}_{tf}$, at time frame t and frequency channel f, the posterior probability of azimuth angle ϕ was computed as

$$P(\phi|\boldsymbol{o}_{tf}) = \frac{p(\boldsymbol{o}_{tf}|\lambda_f^\phi)}{\sum_\phi p(\boldsymbol{o}_{tf}|\lambda_f^\phi)}\,, \tag{2.27}$$

where $p(\boldsymbol{o}_{tf}|\lambda_f^\phi)$ is the likelihood function of GMM, λ_f^ϕ.

The posteriors were then integrated across frequency to produce the probability of azimuth, ϕ, given features $\boldsymbol{o}_t = [\boldsymbol{o}_{t1}^\top, \ldots, \boldsymbol{o}_{t32}^\top]^\top$ of the entire frequency range at time t,

$$P(\phi|\boldsymbol{o}_t) = \frac{\prod_f P(\phi|\boldsymbol{o}_{tf})^{\omega_{tf}}}{P\left(\boldsymbol{o}_t\right)}\,, \tag{2.28}$$

where

$$P(\boldsymbol{o}_t) = \sum\nolimits_\phi \prod\nolimits_f P(\phi|\boldsymbol{o}_{tf})^{\omega_{tf}}\,. \tag{2.29}$$

Assuming that the target sound source was stationary, the frame posteriors were further averaged across time to produce a posterior distribution, $P(\phi)$, of sound-source activity. Here ω_{tf} was introduced as a factor between $[0, 1]$ for selectively weighting the contribution of binaural cues from each time-frequency bin in order to localise the attended target source in the presence of competing sources. When ω_{tf} was zero, the time-frequency was excluded from localisation of the target source. This allowed cues that derived from a frequency channel dominated by the target source to be emphasised. Or conversely, cues that derive from an interfering source could be penalised. Here, top-down information from source models was combined to jointly estimate these localisation weights.

Let λ_s represent the spectral characteristics of a sound source, s, in a set of source models, $s = 1, \ldots, \mathcal{S}$. The set of source models was employed to jointly explain the observed ratemap features. In particular, given the observed log-compressed ratemap feature vector $\boldsymbol{y}_t = [y_{t1}, \ldots, y_{t32}]^\top$ extracted at time frame t from the binaural signals, the goal was to determine whether each feature, y_{tf}, was dominated by the energy of the target source, x_{tf}, or was corrupted by the combined energy of interfering sources, n_{tf}. Under the *log-max* approximation [41] of the interaction function between two acoustic sources, that is, $y_{tf} \approx \max(x_{tf}, n_{tf})$, the localisation weight, ω_{tf}, was defined as the probability of y_{tf} being dominated by x_{tf}, namely,

$$\omega_{tf} = P(x_{tf} = y_{tf}, n_{tf} \leq y_{tf} | \boldsymbol{y}_t, \lambda_x, \lambda_n)\,, \tag{2.30}$$

where λ_x and λ_n are the models for the target and interfering sources, respectively. Here, the source models, λ_s, are represented as GMMs with diagonal covariance matrices. Then, λ_n was built by combining all the source models except that of the target source, that is,

$$p(\boldsymbol{y}_t|\lambda_n) = \sum_{s \neq x} P(s) \sum_{m=1}^{M_s} P(m|\lambda_s) \mathcal{N}\left(\boldsymbol{y}_t; \boldsymbol{\mu}_s^{(m)}, \boldsymbol{\Sigma}_s^{(m)}\right)\,, \tag{2.31}$$

where the prior probabilities of the sound sources, $P(s)$, were assumed to be equiprobable. Alternatively, the above model can be expressed as a large GMM by pooling the Gaussians from all the source models together and multiplying the mixture weights by the corresponding source prior probabilities, so that the resulting mixture weights sum up to one.

Using the expressions for the λ_x and λ_n models in Eq. 2.30, and after some algebraic manipulations, an expression for the localisation weights, ω_{tf}, was derived [16, 34]. See [21] for full details of the derivation.

2.5.3 Evaluation

Binaural audio signals were created by convolving monaural sounds with head-related impulse responses (HRIR) or binaural room-impulse responses (BRIRs). Binaural mixtures of multiple simultaneous sources were created by spatialising each source signal separately before adding them together in each of the two binaural channels. Speech material for the target source was drawn from the GRID corpus [10]. Six types of sounds with various spectro-temporal complexities were used as the interfering sources – as shown in Figure 2.13.

As shown in previous studies [24, 46, 25], multi-condition training (MCT) can increase the robustness of localisation systems in reverberant multi-source conditions. In this study, the localisation models were trained on the binaural MCT features created by mixing a target signal at a specified azimuth with diffuse noise as described in [24]. The source model parameters were estimated from the ratemap features for each source separately, using the EM algorithm. For evaluation, the target source was mixed with one of the interfering sources in a binaural setting. Both target and interfering signals were normalised to the same RMS value prior to spatialisation, and the target source varied in azimuth within the range of $-60\,^\circ$, left, and $60\,^\circ$, right, in $5\,^\circ$ steps. The azimuth of the interferer was randomly selected from the same azimuth range while ensuring an angular distance of at least $10\,^\circ$ between the two competing sources.

The proposed framework was evaluated in the following two scenarios.

(i) The knowledge of the interfering source was assumed to be available *a priori*

(ii) The interfering source was unknown

In the first scenario, the interaction between the target source model and the correct interfering source model for each acoustic mixture was used to estimate the set of localisation weights, ω_{tf}. In the second scenario where the interfering source was unknown, a universal background model (UBM) was created by pooling the Gaussians from all the source models together. The UBM was then used together with the target source model to estimate the localisation weights. A GMM-based baseline localisation system [24] was also evaluated for comparison. This baseline system employed the same localisation models used in the proposed framework, but no top-down knowledge of sound sources was applied.

The gross accuracy rates for localising the target source are shown in Fig. 2.14 for various interferer conditions and reverberant conditions. First, the baseline performance (black bars) shows that the spectral characteristics of interfering sources have an impact on the localisation accuracy. While the 'alarm' source had only a small degrading effect, with gross accuracies above 80 % across all test conditions (both anechoic and reverberant), the presence of other interferers such as the 'telephone ring' had a more detrimental effect. The poor baseline performance in the 'telephone-ring' condition is particularly striking given the narrow-band nature of the sound. However, as shown in Fig. 2.13, the energy of the 'telephone-ring' sound is mostly concentrated in the high frequency range above 2 kHz. It is known that ILDs are more pronounced at high frequencies due to the size of the head compared to the wavelength of incoming sounds. Since the 'telephone ring' dominated

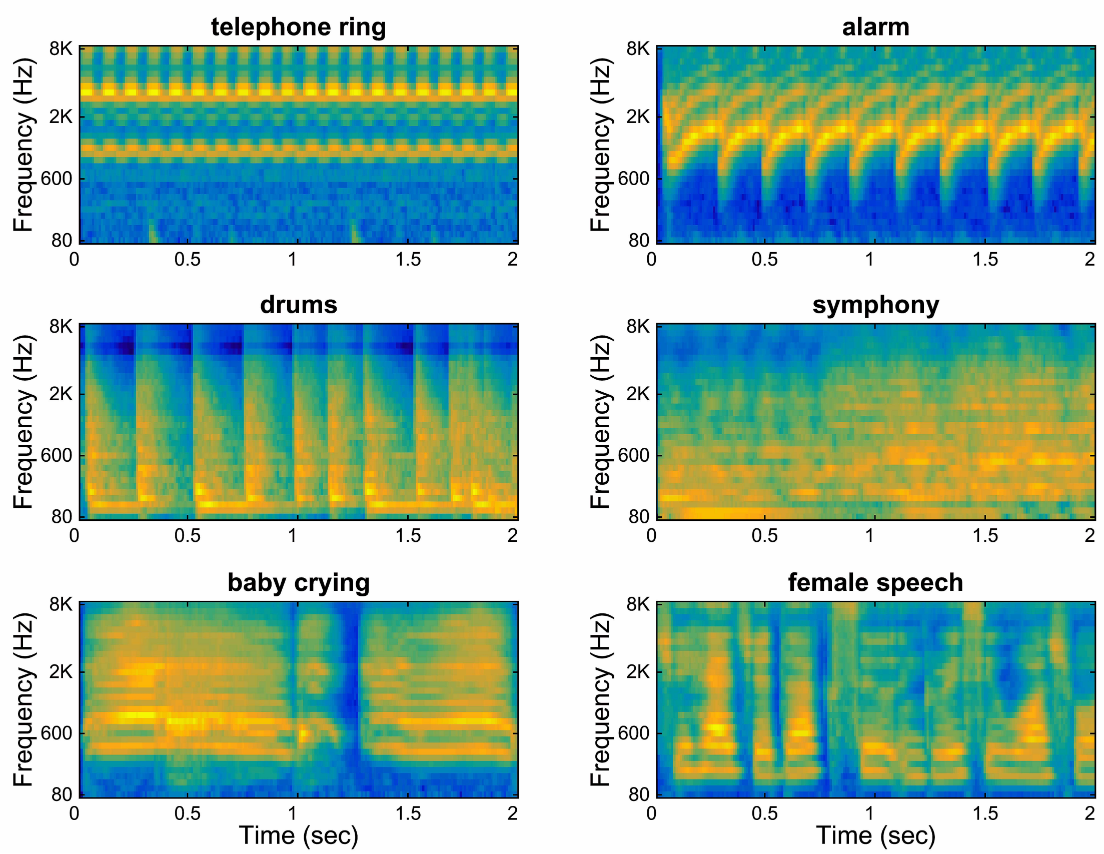


Figure 2.13: Ratemap representations of six interfering sounds

these frequencies across the entire signal duration, only small glimpses were available during which ILDs could be used to localise the target source.

When the spectral characteristics of the active sources in the acoustic scene are available – grey bars in Fig. 2.14 – the proposed approach increased the target-localisation accuracies substantially over the baseline in all the interferer conditions. Interferers with a simpler and more consistent spectral profile were easier to model, and therefore the proposed framework was more effective in such conditions. This is clearly demonstrated by the localisation accuracies in the 'telephone ring' and the 'alarm' conditions, which are close to 100 % – even in the more reverberant rooms.

The localisation accuracy of the proposed system decreased in the more challenging 'symphony' and 'female-speech' conditions. This is likely due to the increased spectro-temporal complexity of the interferers, which becomes more difficult to model statistically. As a result, the localisation weights estimated from source-model interaction were less

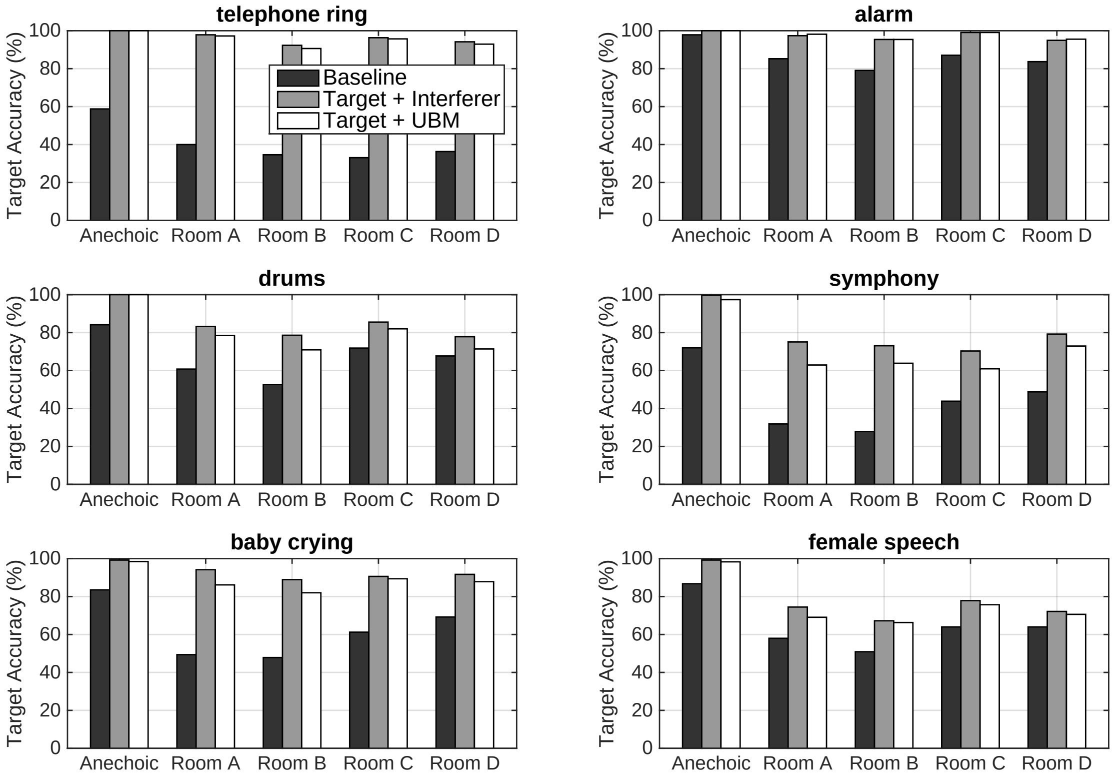


Figure 2.14: Gross accuracies for localising the target source in the presence of various interferers. The 'Baseline' system did not employ top-down source knowledge. The 'Target+Interferer' system employed the interaction between the target model and the correct interferer model for informing the localisation process. The 'Target+UBM' system did not assume knowledge of the interfering source and employed a universal background model instead

reliable.

Comparing the 'Target+Interferer' (grey bars in Fig. 2.14) and the 'Target+UBM' performance (white bars in Fig. 2.14), one can see that with more detailed models of the interfering sources the system produced slightly higher localisation accuracy than with a universal background model – especially for more complex interferers. However, the use of the UBM minimises the assumptions made about the active interfering sources. Such a system is potentially more suitable for an attention-driven model of sound localisation, in which the attended target source may be switched, and the localisation weights can be dynamically recomputed in order to localise the newly attended source.

2.5.4 Discussion

A computational framework for binaural sound localisation was developed that combines top-down and bottom-up information flow. By jointly exploiting top-down knowledge about the source spectral characteristics in the acoustic scene, the system is able to selectively weight binaural cues in order to more reliably localise the attended source. Evaluation using six interfering sources with varying spectro-temporal complexity showed that by exploiting source models in this way, sound localisation performance can be improved substantially under conditions where multiple sources and room reverberation are present. For part of the computations the auditory front end of Two!Ears, (AFE), was applied. The *cognitive* components of of the algorithm will soon be integrated into the Two!Ears system as well.

In the last year of the Two!Ears project, this approach will be fully combined with source identification in order to estimate the identity of the target source that the system 'attends' to. Such an attention-driven model could be used to localise an attended source whose identity is not available *a priori*, that is, a talker that speaks a keyword in an acoustic mixture. Source localisation and source identification could then interact in an ongoing iterative process. Another focus will be cross-modal control. For instance, top-down control within this framework could be driven by the vision system on the Two!Ears mobile robot platform.

2.6 Medial olivo-cochlear feedback and the Precedence Effect

2.6.1 A medial olivo-cochlear (MOC) processor

(The following relates to a1, b2)

A novel processor module was implemented in the Auditory Front-End (AFE) of the Two!Ears system, which attempts to mimic Medial Olivo-Cochlear (MOC) feedback. This so-called MOC processor realises closed-loop-feedback that controls the nonlinear gain at the Dual-Resonance Non-Linear (DRNL) filterbank of AFE – as previously described in deliverable D4.1. The MOC processor builds on internal representations in AFE, which, among other things, simulate the auditory-nerve firing rate. The approach follows work by Clark *et al.* [8], except that the auditory-nerve processing stage was simplified by using the ratemap processor of AFE. Figure 2.15 describes the structure and operation of the MOC processor in conjunction with AFE processing stages employed.

The MOC processor initially takes the ratemap representation as reflexive input. This is then converted to attenuation factors for each individual frequency channel. The control of the rate-to-attenuation conversion is realised via internal parameters of the processor. These can be set in accordance with physiological findings that reveal the relationship between input-signal level and MOC efferent activity, for example, those from experiments with cats reported by Liberman [20] – see Fig. 5 of that paper. The parameters are configurable by the user when requesting the internal representation with DRNL and MOC to be activated.

Further, the reflective feedback path is controllable by means of additional attenuation factors, particularly, ipsi- and contralateral ones, depending on the specific application. These factors are accessible internally, that is, within AFE, as well as externally, for example, from the blackboard system. The factors can be set to arbitrary values. In this way, a cognitive processor can be enabled to control the amount of feedback desired.

Overall, the MOC processor, at its current stage of development, provides a comprehensive testbed for users to simulate currently available findings or to even manipulate the MOC feedback for any further relevant investigation. As stated in D4.1, the lack of consistent physiological findings with repect to the physiological role of MOC feedback currently restricts the capabilities of the model and prevents it from being applied to functionally improve the Two!Ears system. However, the nature of the model, particularly, the fact that it inherits the modularity of the Two!Ears framework, enables easy modification of its operational structure. Thus, if necessary, it can readily be activated, in particular, when new findings become avaliable.

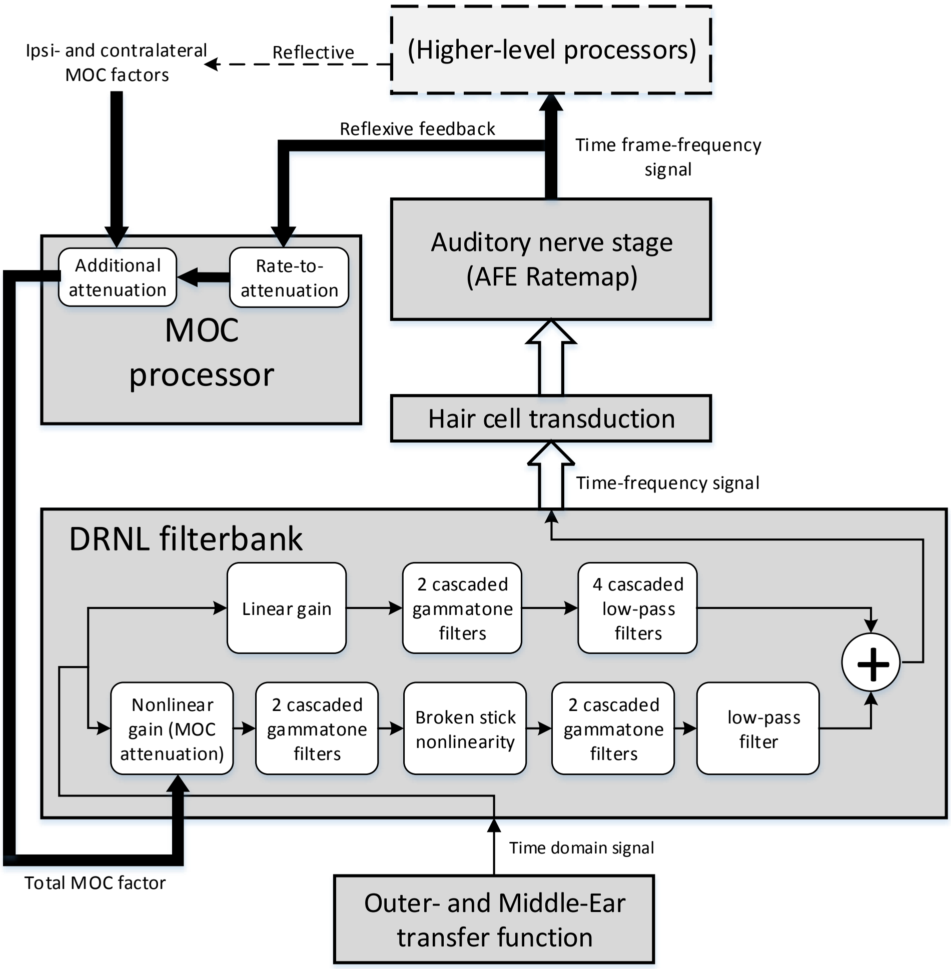

Ipsi- and contralateral MOC factors
Reflective
(Higher-level processors)
Reflexive feedback
Time frame-frequency signal
Additional attenuation
Rate-to-attenuation
MOC processor
Auditory nerve stage (AFE Ratemap)
Hair cell transduction
Time-frequency signal
DRNL filterbank
Linear gain
2 cascaded gammatone filters
4 cascaded low-pass filters
+
Nonlinear gain (MOC attenuation)
2 cascaded gammatone filters
Broken stick nonlinearity
2 cascaded gammatone filters
low-pass filter
Time domain signal
Total MOC factor
Outer- and Middle-Ear transfer function

Figure 2.15: MOC processor design within the auditory front-end framework. The processor realises reflexive feedback from the output of the auditory-nerve stage and accepts reflective control from higher-level cognitive stages. The output is applied to the nonlinear path of the DRNL filterbank.

2.6.2 Precedence-Effect processor

Further, an AFE processor developed and intergrated in AFE to simulate the *Precedence Effect*. The hereby applied algorithm is based on work of Braasch [5]. The model fundamentally detects and removes reflections from an input signal by means of autocorrelation followed by deconvolution. After these processes, it derives the interaural time and level differences, ITD and ILD, from these lag-removed signals, represented in the auditory periphery domain.

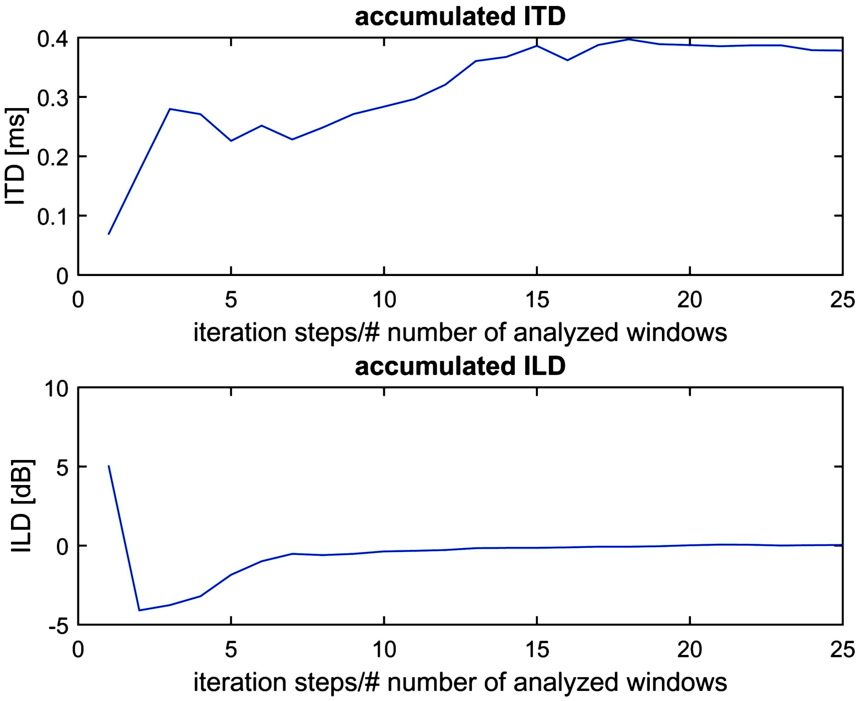


Figure 2.16: Precedence-Effect-model demonstration. The input signal is a 800-Hz-wide bandpass noise of 400 ms length, centered at 500 Hz, mixed with a reflection of a 2-ms delay, and made binaural with an ITD of 0.4 ms and an ILD of 0 dB. The instantaneous ITD/ILD estimation is performed within signal chunks of 20 ms

The initial version of the Precedence-Effect model was made available for monaural analysis (lag detection and removal) via the *Auditory-Modeling-Toolbox* software framework (AMT). The model published there makes use of the complete signal instead of using time-frame chunks of it. This basic model was now developed further and integrated into the Two!Ears framework as *PrecedenceProc*. To this end, a number of modifications were made. These modifications allow for, (a), object-based modular operation in conjunction with the available AFE processors and, (b), chunk-based processing for lag detection and removal instead of batch analysis of complete monaural signals. The architecture of the integrated model is described in [5] – see Fig. 9 of that paper. A difference to the original model is, however, that the 'Hair-cell simulation' and the 'Halfwave Reconstruction' stages were removed. The original model contains a halfwave-reconstruction stage to enable easy connection to physiologically motivated inner-hair-cell models. Since, in the inner-hair-cell processors, halfwave reconstruction is effectively the reverse operation of halfwave rectification. Hence, cascading of the two is equivalent to bypassing them as a whole. Both were thus removed in our case.

The Precedence-Effect processor, as now integrated into the Two!Ears framework, takes as input binaural signal chunks from the Gammatone filterbanks. Then, for each chunk, a pair of ITD and ILD values is calculated by integrating the ITDs and ILDs across frequency channels according to the weighted-image model of Stern [37] and via amplitude-weighted summation. Since these ITD/ILD calculation methods are different from those typically used in the AFE framework, the ITD and ILD processors in the AFE are not connected to the Precedence-Effect processor. Instead the ITD/ILD calculation steps are coded separately.

Figure 2.16 shows the output of a demonstration using the Two!Ears Precedence-Effect processor. The input signal is a 800-Hz-wide bandpass noise of 400 ms length, centered at 500 Hz, mixed with a reflection having a 2-ms delay, and made binaural with an ITD of 0.4 ms and an ILD of 0 dB. During the processing, a windowed chunk of 20–ms length is used as input. It can be seen that after some initial fuzziness, the processor estimates the intended ITD and ILD values – with increasing precision as more chunks are analysed.

As with the MOC feedback processor, the modular and feedback-enabling nature of the AFE framework makes it possible for the user to request the various ITD and ILD representations using the conventional AFE models, and/or using the Precedence-Effect model – depending on the specific application. Following decisions at cognitive level, a request to switch the Precedence-Effect model on or off can be issued, for example, from the blackboard system.

2.7 Sound-type classification and feature selection – potential roles of feedback

(The following relates to b6)

Here it was investigated whether context information, that is, the knowledge of certain aspects of auditory scenes, can help the Two!Ears system to enhance the performance in terms of sound-source identification. To this end, a selection of everyday sounds from the NIGENS database was used to generate simple auditory scenes using the *Auditory Machine-Learning Training and Testing Pipeline* (AMLTTP) – a software package based on Two!Ears's Auditory Frontend (AFE). The scenes were composed from, (1), 'dry' sounds overlaid with ambient white noise of different strength (SNR) and, (2), target sounds overlaid with simultaneously played distractor sounds from a point source at different azimuth. The composed scenes were then processed by AFE via AMLTTP in order to generate a large set of candidate features. State-of-the-art data-driven feature-selection methods, namely, *Least Absolute Shrinkage and Selection Operator Techniques* (LASSO), were combined with cassifiers based on *Support-Vector-Machines* (SVM), to tackle following questions.

1. Does the feature-selection step improve the performance of the identification-knowledge sources?
2. Are different sets of features selected for the different conditions – SNR, azimuth difference of target versus distractor?
3. Can improved classification results be achieved by adapting feature sets and/or classifiers to the particular condition?

Sufficient evidence for an improvement of classification by condition-dependent event experts would then serve as the justification for the implementation of feedback loops that select feature sets and/or classifiers based on estimates of the current condition at runtime – such as the directions of sound sources and/or the signal-to-noise ratio (SNR). The following results were achieved.

1. On average, classification performance did not improve significantly for the stimulus set used when SVM-based classifiers were applied to preselected feature sets, rather than being applied to all candidate features. However, computational costs were much less when feature selection was applied beforehand, because it allowed for a drastic reduction of the number of features to be computed for classification.
2. For datasets (1) & (2), the application of feature-selection methods provided strong evidence for the advantage of sound-class-specific feature sets – as one would have

expected. Also, the profile of the selected feature set changed with the condition – dataset (1).

3. For both datasets (1) & (2), classifier adaptation to the particular condition, such as the azimuth of the target and/or the SNR of the data, lead to performance improvements on average.

It can be concluded that for the simple tasks that were considered, evidence pointed to a potential improvement of sound-type-classification performance when feedback loops are applied for feature-set tuning and condition-based-classifier selection. This holds in particular for the case of target sounds played from different directions when proper turns of the head of the Two!Ears robotic platform are induced.

2.7.1 Datasets and preprocessing

The NIGENS database currently consists of twelve classes of everyday sounds. From these, eleven *one-against-all* binary sound-classification tasks were investigated, such that one particular sound class had to be classified against the rest.

The examples for classification training were generated from blocks (time windows) of 500 ms. Two sets of features were generated for each block. The monaural feature set (average over the two channels) comprises a total of 1082 features that were extracted from the output of Two!Ears's Auditory Frontend (AFE), of which 154 were spectral features, 176 ratemap features, 576 amplitude-modulation features, and 176 onset-strength features. The binaural feature set, as computed separately for the left and right channel, had a size of 2164.

2.7.2 Methods

Feature selection

For feature selection, the LASSO method was employed. LASSO is an embedded feature-selection method that is based on a linear (logistic) regression model. It applies a penalty, L1, to the regression coefficients, thus shrinking many of them to zero. The strength of the L1 regularization is controlled by a hyperparameter, λ. The value of λ was adjusted by conducting a five-fold cross-validation on the training set and choosing the value with the best cross-validation performance from the values of 100 candidates taken from the regularization path. Any features that had a non-zero regression coefficient at this value of λ were then used as input features for the SVM (scheme fs1). Alternatively the highest value of those λs was chosen, for which the cross-validation performance was greater than or equal to the difference between the maximum cross-validation performance over all values for λ and its standard deviation (scheme fs3).

Classification

For classification, a linear C-Support-Vector Machine (C-SVM) was used. SVMs are classification models with associated learning algorithms derived in the context of statistical learning theory. Parameters were adjusted by maximizing the margin of a hyperplane separating the two classes that could be related to a bound on the generalization performance of the classifier. If the training data are not linearly separable, so-called slack variables need to be introduced that allow for violations of the margin. The sum of these slack variables served as a penalty term and was weighted by a hyperparameter, C. Here, C was adjusted via four-fold cross-validation on the training set within the parameter set of $10^{-8}, 10^{-7}, \ldots, 10^{-1}$. Classification performance was always evaluated on a held-out test set.

Evaluation

Data were split into a training set and a test set, where the training set was used to construct the classification model, and the test set was used to evaluate the prediction performance. Performance was measured with the variant

$$\text{performance} = 1 - \sqrt{((1 - \text{sensitivity})^2 + (1 - \text{specificity})^2)/2)} \tag{2.32}$$

of the balanced accuracy that combines sensitivity and specificity and favors similar values for both.

2.7.3 Results

In a first experiment, the performance of the C-SVM, trained and tested on a complete set of features, was compared with the performance of a two-stage procedure, where feature selection using LASSO method was applied before the selected features were used as input for the C-SVM. Figure 2.17 shows the results for dataset (1), where the different sound classes – denoted by color – were superimposed with ambient white noise of different SNR – denoted by marker size. All symbols lie close to the diagonal, thus indicating that the data-driven feature-selection procedures do not lead to a significant increase in the performance of sound classification. Similar results were obtained for dataset (2).

Feature selection, however, allows for a drastic reduction of the number of features without

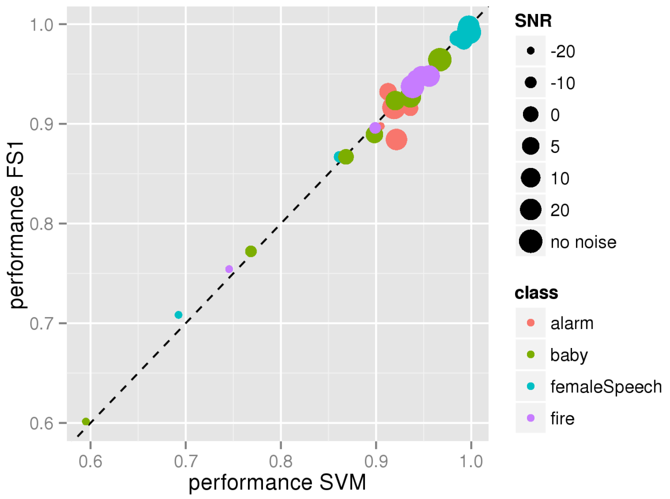


Figure 2.17: Classification performance of an SVM trained on the whole monaural feature set (abscissa) in comparison to the performance of an SVM trained on a reduced (monaural) feature set selected by LASSO (scheme fs1, ordinate). Different colors correspond to the different classes ‘alarm’, ‘crying baby’, ‘female speech’ and ‘fire’. Different marker sizes indicate different values of the signal-to-noise ratio

loss of performance. Depending on the sound class, the reduced feature sets range from a few features – reduction to 1 % and less – up to a few hundred features – reduction to 30%. Most importantly, it became apparent that application of feature selection leads to drastically reduced computing times for both training and prediction – compare deliverable D3.4 for details. In particular, the reduction in classification time could prove to be a crucial factor in a real-time operation of the Two!Ears system.

It was further investigated whether the selected feature sets change with sound class and condition – dataset (1). Candidate features were grouped into the following base groups: spectral features, ratemap features, amplitude modulation features, and onset-strength features. Then the ‘impact’ of features selected from each group was computed. As a measure, the normalized sum of the absolute weights of the logistic regression model was applied to assess the influence of a feature group on the classification result. Every sound class leads to a different profile, that is, to a different set of values across the four groups (data not shown). This was taken as an indication that classifiers rely on different information when classifying sounds from different classes. In addition, there was a dependency of the feature-group impacts on the SNR – see Fig. 2.18.

With increasing noise versus decreasing signal strength, the onset-strength and ratemap features became less important, while the impact values of the amplitude modulation features increased.

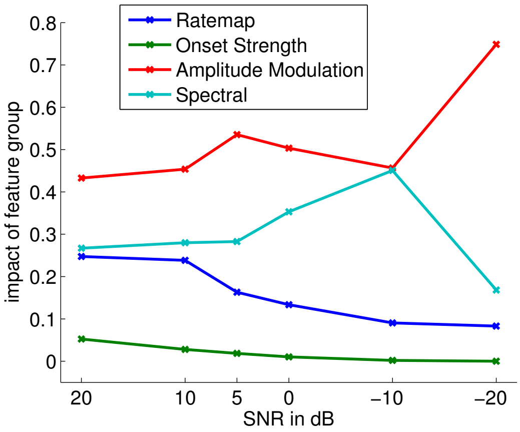


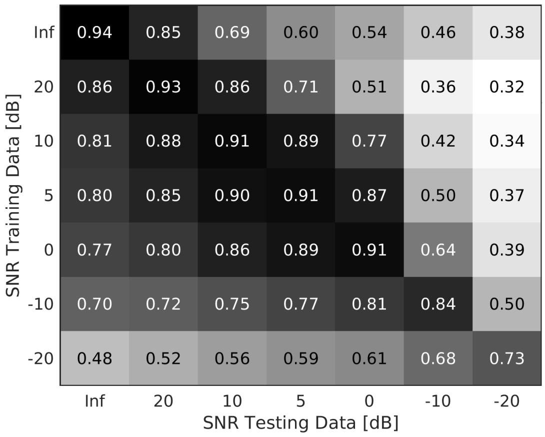
Figure 2.18: Impact of features (normalized sum of the absolute logistic regression weights) selected from four different categories as a function of the SNR for dataset (1). Values were averages over all eleven classes from the NIGENS database. Features were selected by use of scheme fs3

In a second experiment, it was investigated how well the two-stage method – LASSO for feature selection followed by SVM for classification – performed under iso-testing, that is, testing at the same noise condition that was used for training, versus cross-testing conditions, namely, testing at a different noise condition than used for training.

Figure 2.19 shows a matrix the elements of which contain the test performance of the classifier averaged over all eleven sound types for different combinations of training SNR (rows) and testing SNR (columns). Diagonal elements correspond to the iso-testing condition, off-diagonal elements to the cross-testing conditions. The best performance was always obtained for the iso-conditions. For a given SNR of the test set, the average generalization performance of the classifier decreased with increasing difference to the SNR of the training condition. This decrease in performance became stronger with increasing SNR of the test set. Especially at high noise levels, the classifiers needed to be adapted to the particular condition in order to generalize well.

This suggests that sound-type identification by the Two!Ears system could be improved by providing information about the likely SNR of the observed data. This information could then be used – in a feedback fashion – to select a feature-selection and classification model that was trained under similar conditions. An alternative to the use of such specialized experts is the use of multi-conditional training – compare Chaps. 3 & 4 of deliverable D3.4. The classifiers resulting from multi-conditional training, however, perform worse on average than the single-condition experts.

In a third experiment it was investigated whether the use of binaural feature sets would lead to improved classification performance as compared to monaural feature sets. The comparison was performed for tasks where target sounds were overlaid with distractor



	Inf	20	10	5	0	-10	-20
Inf	0.94	0.85	0.69	0.60	0.54	0.46	0.38
20	0.86	0.93	0.86	0.71	0.51	0.36	0.32
10	0.81	0.88	0.91	0.89	0.77	0.42	0.34
5	0.80	0.85	0.90	0.91	0.87	0.50	0.37
0	0.77	0.80	0.86	0.89	0.91	0.64	0.39
-10	0.70	0.72	0.75	0.77	0.81	0.84	0.50
-20	0.48	0.52	0.56	0.59	0.61	0.68	0.73

Figure 2.19: Classification performance averaged over all eleven sound classes (monaural feature set) for the iso- and cross-testing conditions. Vertical and horizontal axes correspond to the SNRs used for training and test. Matrix entries denote the corresponding prediction performances. In addition, performance values are visualized by a grey level encoding from black (good performance) to white (bad performance).

sounds, both played from different directions – dataset (2). Tasks with target- and distractor-sound sources were used that were located at $-45\,^{\circ}, +45\,^{\circ}$ and $-90\circ, +90\,^{\circ}$ azimuth and had different SNRs. Figure 2.20 shows a summary box-plot of the difference in performance found between predictors trained with monaural and binaural features for different SNRs. The binaural feature set lead to a better performance when compared to the monaural feature set, specifically for low values of the SNR. The stronger the distractor source became, the more advantageous it was to use binaural information from both ear signals.

Figure 2.21 shows the performance of classifiers trained by using the binaural feature set for specific values of the azimuth of the target source but without distractors being present. For the iso-testing condition, there was no significant change in performance as a function of the azimuth of the sound source – Fig. 2.21a. Performance values, however, changed for the cross-testing condition – Fig. 2.21b. Here, performance decreased with increasing difference of the azimuth values used for training and evaluation[2]. This implies that either the head should be turned to adjust the azimuth of the sound source to the azimuth which was used for training the classifier or, that the model has to be switched dynamically to the one that was trained at the actual azimuth, both of which can be implemented via feedback loops.

[2] Because tests were conducted for sound sources located in the right hemisphere, performance increased again for azimuth differences higher then $90\,^{\circ}$

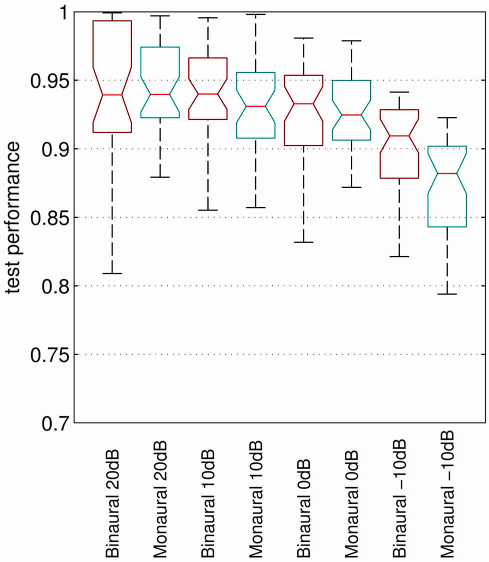
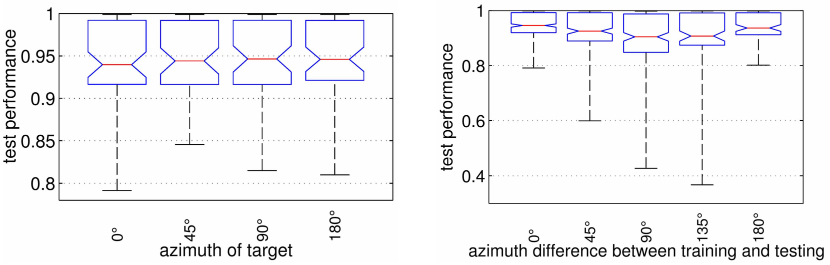


Figure 2.20: Classification performance of models trained on dataset (2) using both monaural and binaural feature sets at different SNRs, with the conditions of target- and distractor-source azimuth at $-45\,°, +45\,°$ and $-90\,°, +90\,°$. The boxplots summarize the results across both conditions for the four classes 'alarm', 'crying baby', 'female speech', and 'fire', as well as the results from the different classification schemes



(a) dependence on target azimuth

(b) performance degradation with difference of training and testing azimuth

Figure 2.21: Performance of classifiers trained with the binaural feature set for specific values of the azimuth of the target source but without distractors being present. The boxplots summarize the results across the four classes and different classification schemes. (a) Test performance as a function of the azimuth of the target source (iso-testing). (b) Test performance as a function of the difference between the azimuths of the target in the training and testing phases, that is, iso- versus cross-testing

2.8 Sensorimotor-cue processing for sensorimotor feedback control

(The following relates to b7)

2.8.1 Introduction

The sensorimotor level constitutes the lowest layer both in the TWO!EARS computational model and in the deployed robotics architecture. Situated on the top of instrumentation, it is constituted of reflexive behavior involving perception and/or motion. These respective processors must run under severe time and communication constraints and do not entail any reflective ability, such as decision or cognition.

Cues coming from the low-level processing of the sensorimotor flow can constitute meaningful input to the blackboard, thus supporting decisions in the TWO!EARS system. Actually, the involved processors can themselves be viewed as experts of the model. On this basis, dedicated sensorimotor functions can be triggered that have been implemented on further experts.

One robotics example is binaural 'active' sound-source localization which, through the incorporation of motion, enables the disambiguation of front from back and the recovery of source range [26]. Work has been performed with regard to a single-source-localization strategy organized into three layers – as depicted in Fig. 2.22 [7]. Stage A implements the maximum likelihood estimation of the source azimuth and the information-theoretic detection of its activity from the short-term channel-time-frequency decomposition of the binaural stream [33]. Stage B assimilates these azimuths over time and combines them with

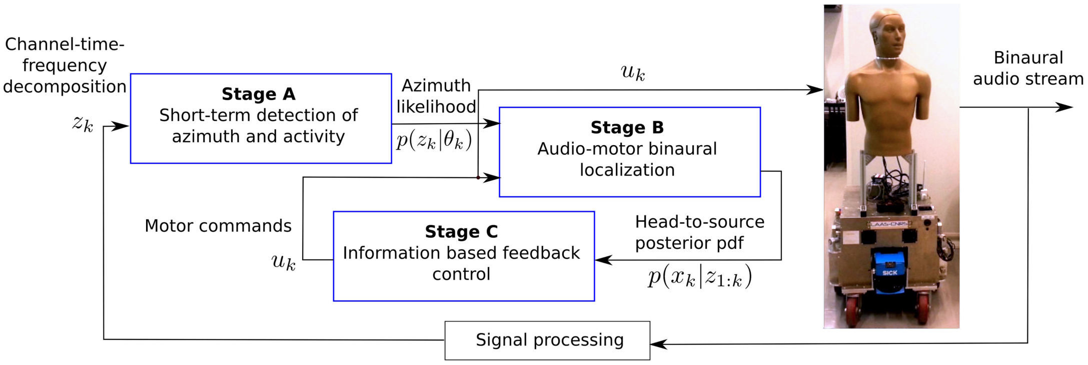

Channel-time-frequency decomposition
z_k
Stage A
Short-term detection of azimuth and activity
Azimuth likelihood
$p(z_k|\theta_k)$
u_k
Stage B
Audio-motor binaural localization
Binaural audio stream
Motor commands
u_k
Stage C
Information based feedback control
Head-to-source posterior pdf
$p(x_k|z_{1:k})$
Signal processing

Figure 2.22: Three-stage active binaural localization

the motor commands into a stochastic filter, leading to the posterior probability-density function (pdf) of the head-to-source situation [31]. Stage C provides a feedback controller, which, on the basis of output from Stage B, can move the head so as to improve the quality of the localization, that is, of the output from Stage B.

Stage A has also been extended to the multiple source case [30], and Stage B can cope with a moving and/or intermittent source [32], but this is not considered here. This whole active binaural localization, though involving solely the sensorimotor layer, perfectly fits within the Two!Ears blackboard-based architecture [3].

Since the goal of the comprehensive Two!Ears model extends far beyond active localization, the control signals computed in Stage C may not be applied to the binaural robot – in contrary to the diagram in Fig. 2.22. In fact, it is up to the whole blackboard-based decision process to make an adequate use of these signals, for example, to trigger a reflexive information-based control on a short-time duration or to combine them with other information in order to synthesize effective motor commands for the robot.

This chapter outlines first theoretical considerations and results towards feedback control of sensor motion for reducing localization uncertainty in Stage C. Deliverable D4.1 reported literature on active information-based sensorimotor feedback. When restricting to the auditory modality, it included audio-motor localization, audio simultaneous localization and mapping (SLAM), and planned (reflective) motions to improve speech recognition or sound-localization accuracy. It may be briefly recalled at this point that within the 'exploration problem' in robotics, robots also move autonomously so as to maximize their knowledge about the world. SLAM has been extended in such a way that robots move into the direction of maximum local information improvement [38]. Therein, the control scheme extracts the information about the state variables using the concepts of Shannon entropy and mutual information [11], and, in the end, maximizes a 'size' criterion such as the determinant or trace of the inverse of the *one-step ahead posterior-state-covariance matrix* [4].

Other information-theoretic controllers have been applied to information retrieval on some targets [17], robot guidance towards areas of maximum uncertainty [18], control of a robot-mounted camera to optimize depth estimation [14], or sensor-parameter setting (such as for zoom or attitude) for scene analysis [12, 35]. As already mentioned in D4.1, auditory information-based control is sparser. Motion planning was proposed in [19] to improve speech recognition from a monaural robot. Sound localization was improved in [23] by moving microphones deployed in the environment. In a noticeable recent work [42], a robot equipped with a microphone array has been controlled to improve occupancy-grid-based source localization by using dynamic programming.

2.8.2 Overview of Stage A and Stage B

This section briefly recalls the azimuth estimation for a single source [33] and the incorporation of head motion [31] so as to get a Gaussian-mixture approximation of the posterior pdf of the head-to-source situation. In the sequel, random variables/processes and corresponding samples are written using similar lower-case letters.

Short-term extraction of directional cues

The left and right microphones are termed R_1 and R_2. The interaural transfer function is known over an adequate range of source azimuth and frequencies. The frame, $\mathcal{F} = (O, \boldsymbol{e_x}, \boldsymbol{e_y}, \boldsymbol{e_z})$, is attached to the head, with $\boldsymbol{R_1O} = \boldsymbol{OR_2}$.

R_1, R_2 and the pointwise emitter, E, lie on a common horizontal plane defined by $\boldsymbol{e_y}, \boldsymbol{e_z}$, where $\boldsymbol{e_y} = \frac{\boldsymbol{R_2R_1}}{\|\boldsymbol{R_2R_1}\|}$, and $\boldsymbol{e_z}$ is oriented towards boresight, so that $\boldsymbol{e_x}$ points downwards – Fig. 2.23. The source and sensor noises are modeled as random processes satisfying reasonable hypotheses, such as Gaussianity, zero-mean, band-limited, 'local stationarity'.

From [33], on the basis of the channel-time-frequency decomposition, z_k, of the binaural signal on a sliding window ending at time k, the short-term-maximum likelihood, $\hat{\theta}_k$, of the source azimuth, θ_k, comes as the argmax of a 'pseudo likelihood', $p(z_k|\theta_k)$. This pseudo likelihood is obtained by replacing in the genuine likelihood of the unknown variables the most likely spectral parameters of the source as a function of its azimuth – thanks to a notable separation property.

Fusion of audio information with motor commands

A discrete-time stochastic state-space equation is set up, uniting the velocity-control vector, $u_k \in \mathbb{R}^3$ (2 translations and 1 rotation), of the head to the head-to-source situation, $x_k \in \mathbb{R}^2$ – see Fig. 2.23. A theoretically sound Gaussian-mixture square-root-unscented Kalman filter (GMsrUKF) is defined so as to incorporate the above pseudo likelihood, $p(z_k|\theta_k)$, where θ_k comes as a static function of x_k [31] as well as to compute a Gaussian mixture approximation of the posterior pdf as follows,

$$p(x_k|z_{1:k}) = \sum_{i=1}^{I_k} w_k^i \mathcal{N}(x_k; \hat{x}_{k|k}^i, P_{k|k}^i), \tag{2.33}$$

where $(w_k^i, \hat{x}_{k|k}^i, P_{k|k}^i)$ are the weight, mean, and covariance of each hypothesis.

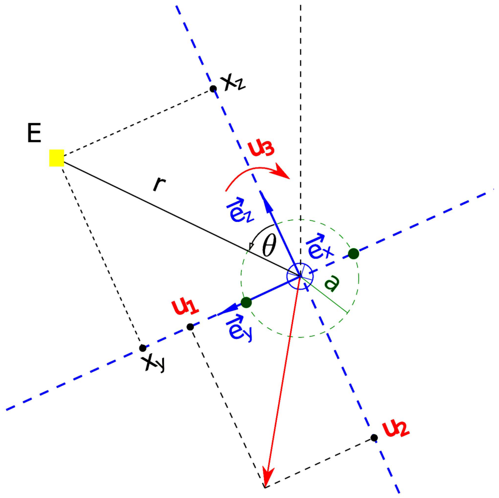


Figure 2.23: Problem geometry

Contrarily to several particle filters, a self-initialization as well as posterior covariance consistency is ensured, so that front and back are disambiguated, and both range and azimuth are faithfully recovered.

2.8.3 Towards information-based sensorimotor feedback

An information measure can be defined from the posterior pdf (Eq. 2.33) at time k, which captures all the information on the head-to-source situation held in the measurements.

The *one-step-ahead-control problem* is studied, consisting in determining the control vector, u_k^*, such that the information in $p(x_{k+1}|z_{1:k+1})$, averaged over the (unknown) possible values of the next measurement, z_{k+1}, is maximized. Two simplifications make the problem tractable. First, $p(x_k|z_{1:k})$ is reduced to a single Gaussian pdf, $\mathcal{N}(x_k; \hat{x}_{k|k}, P_{k|k})$, for example, by keeping its most probable hypothesis or by computing its moment-matched approximation. Secondly, in order to define u_k^*, the next channel-time-frequency decomposition, z_{k+1}, is traded for a scalar vector, y_{k+1}, satisfying a closed-form measurement equation, $y_{k+1} = g(x_{k+1}) + v_{k+1}$, with v being the measurement noise.

So far, the Woodworth-Schlosberg formula for interaural-time-difference approximation over a spherical head [1] has been selected for $g(.)$, in order to guide the exploration. u_k^* is then defined on the basis of $\mathcal{N}(x_k; \hat{x}_{k|k}, P_{k|k})$ and on the above measurement equation. Once this control signal is applied, the next state posterior pdf, $p(x_{k+1}|z_{1:k+1})$, is computed from Eq. 2.33 and the next likelihood, $p(z_{k+1}|\theta_{k+1})$, by using the GMsrUKF described in

Sec. 2.8.2. Then, the whole process is repeated to determine u^*_{k+1}.

To ease the presentation of the synthesis of u^*_k's, henceforth, it is set to $p(x_k|y_{1:k}) = \mathcal{N}(x_k; \hat{x}_{k|k}, P_{k|k})$ – with a slight notation misuse.

Information measures and control input

Let w, x be two random variables with the pdfs $p(w)$ and $p(x)$. The differential entropy, $h(x)$ of x, embodies its uncertainty in that the lower $h(x)$ the higher the information content in x. The mutual information, $I(w, x)$ (≥ 0 by definition), measures the amount of information that w contains about x [11]. They are defined by

$$\begin{aligned} h(x) &= -\int p(x)\log p(x)dx, & (2.34)\\ I(w,x) &= \iint p(w,x)\log\frac{p(w,x)}{p(w)p(x)}dw\,dx\,. & (2.35)\end{aligned}$$

If w and x are conditioned on the event such that a random variable, v, takes a given value, then the entropies/information are denoted by $h(w|v)$, $h(x|v)$ and $I(w, x|v)$. The following rule, somewhat similar to [22], holds.

Theorem 1 *Decomposing the negative logarithm of the posterior pdf, $p(x_{k+1}|y_{1:k+1})$, as*

$$-\log p(x_{k+1}|y_{1:k+1}) = -\log p(x_{k+1}|y_{1:k}) - \log\left(\frac{p(y_{k+1}|x_{k+1})}{p(y_{k+1}|y_{1:k})}\right), \qquad (2.36)$$

and taking its expectation conditioned on $y_{1:k}$, which involves the joint pdf, $p(x_{k+1}; y_{k+1}|y_{1:k})$, leads to

$$\begin{aligned} \mathsf{E}_{y_{k+1}}\{h(x_{k+1}|y_{1:k+1})\} &= h(x_{k+1}|y_{1:k}) - I(x_{k+1}; y_{k+1}|y_{1:k}) & (2.37)\\ \mathsf{E}_{x_{k+1}}\{h(y_{k+1}|x_{k+1})\} &= h(y_{k+1}|y_{1:k}) - I(x_{k+1}; y_{k+1}|y_{1:k}), & (2.38)\end{aligned}$$

with $h(x_{k+1}|y_{1:k+1})$, $h(x_{k+1}|y_{1:k})$, $h(y_{k+1}|x_{k+1})$, $h(y_{k+1}|y_{1:k})$, the entropies of the next filtered-state pdf (head-to-source situation), the next predicted-state pdf, the observation law, the next predicted-measurement pdf, and $I(x_{k+1}; y_{k+1}|y_{1:k})$, the mutual information of the next state, and measurement conditioned on the sequence of measurements up to current time.

In view of the mutual information positivity, the inequality $\mathsf{E}_{y_{k+1}}\{h(x_{k+1}|y_{1:k+1})\} \leq h(x_{k+1}|y_{1:k})$ holds, which highlights the entropy reduction brought by the measurement process. Thus, given $p(x_{k+1}|y_{1:k})$, minimizing $\mathsf{E}_{y_{k+1}}\{h(x_{k+1}|y_{1:k+1})\}$ boils down to maximizing $I(x_{k+1}; y_{k+1}|y_{1:k})$.

The entropy of a normal distribution comes as the following increasing affine function of the log-determinant of its covariance matrix [11],

$$h(x_{k+1}|y_{1:k+1}) = \frac{1}{2}\log\left[(2\pi e)^n |P_{k+1|k+1}|\right] . \tag{2.39}$$

Some important simplifications on Eqs.,2.37–2.38 follow. First, $h(y_{k+1}|x_{k+1})$ only depends on the variance of the noise, v_{k+1}. Second, the (nonlinear) Kalman equations that could be used to assimilate the measurement, y_{k+1}, for exploration show that $h(x_{k+1}|y_{1:k+1})$ does not depend on y_{k+1}. Finally, as the prior state dynamics turns to a rigid motion of the head, the entropy, $h(x_{k+1}|y_{1:k})$, of the next predicted state pdf is equal to $h(x_k|y_{1:k})$ when the dynamic noise is neglected, whatever the applied control signal, u_k. So, the following theorem holds.

Theorem 2 *Finding a control input, u_k^*, which minimizes the entropy, $h(x_{k+1}|y_{1:k+1})$, of the next filtered state pdf (or, equivalently, its expected value over y_{k+1}) is equivalent to maximize the mutual information, $I(x_{k+1}; y_{k+1}|y_{1:k})$, of the next predicted state and measurement conditioned on the sequence of measurements up to current time or, equivalently, to maximize the entropy, $h(y_{k+1}|y_{1:k})$, of the next predicted measurement pdf, that is,*

$$\begin{aligned} u_k^* = \arg\min_{u_k} \mathsf{E}_{y_{k+1}}\{h(x_{k+1}|y_{1:k+1})\} &= \arg\max_{u_k} I(x_{k+1}; y_{k+1}|y_{1:k}) & (2.40)\\ &= \arg\max_{u_k} h(y_{k+1}|y_{1:k})\,. & (2.41)\end{aligned}$$

The Kalman-filter equations entail the approximations $\hat{y}_{k+1|k}$ and $S_{k+1|k}$ of the predicted measurement mean and covariance. Considering that $p(y_{k+1}|y_{1:k})$ is approximated by the Gaussian pdf, $\mathcal{N}(y_{k+1}; \hat{y}_{k+1|k}, S_{k+1|k})$, the entropy, $h(y_{k+1}|y_{1:k})$, can be rewritten as an increasing affine function of the log-determinant of $S_{k+1|k}$, in the vein of Eq. 2.39.

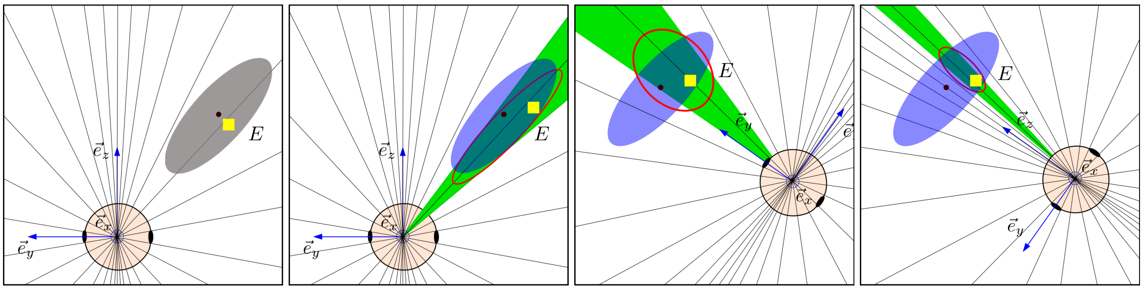
Geometric interpretation

Theorem 2 can be interpreted geometrically. Given a genuine head-to-source situation (Fig. 2.24 a), the 2–D Gaussian approximation of the next filtered-state pdf resulting from the fusion of the next predicted-state pdf with the measurement is represented for various positions. These are, when the sensor is still (Fig. 2.24 b), when after a sensor motion the interaural axis, supported by $\boldsymbol{e_y}$ (resp. the boresight direction, supported by $\boldsymbol{e_z}$), are parallel to the small axis of the confidence ellipse associated to the predicted-state pdf – see Figs. 2.24 b), resp. 2.24 c–d).

Importantly, the Woodworth iso-ITDs are not uniformly distributed along the azimuths.

They are more concentrated along the direction of $\boldsymbol{e_z}$, which defines the auditive fovea, while they are sparser along the interaural axis $\boldsymbol{e_y}$.

The variance of the predicted measurement is low when the ellipse is intersected by a small number of iso-ITDs (Fig. 2.24.b-c). In this case, the measurement uncertainty due to noise corresponds to a wide spatial sector. Consequently, the measurement update cannot significantly improve the information in the filtered state pdf. The more iso-ITDs intersect the ellipse associated to the predicted state pdf, the higher the variance of the predicted measurement. For instance, when the small axis of this ellipse is parallel to the auditive fovea (Fig. 2.24.d) or when the head gets closer to the source, the measurement uncertainty due to noise corresponds to a narrow cone. Then, the fusion of the predicted state pdf and of the measurement results in a strong increase of the information in the filtered state pdf.



(a) (b) (c) (d)

Figure 2.24: The blue frame, $\mathcal{F} : (O, \boldsymbol{e_x}, \boldsymbol{e_y}, \boldsymbol{e_z})$, is attached to the binaural head, represented by its ears. The genuine position of the sound source is pointed at with a yellow square. The grey confidence ellipse is related to the posterior pdf of the current head-to-source situation. The measurement space is materialized by the Woodworth iso-ITDs. The blue ellipses are associated with the next-predicted-state pdf – after applying the velocity command to the head, which is zero in (b). The green sector/cone describes the spatial uncertainty due to measurement noise. The red ellipse sketches the confidence ellipsoid associated to the next-filtered-state pdf, after incorporation of the Woodworth ITD for the source position

Feedback control by gradient-ascent strategy

As the robot head undergoes a rigid body motion, the problem is reduced to find, from the head-to-source situation at time k characterized by $p(x_k|y_{1:k}) = \mathcal{N}(x_k; \hat{x}_{k|k}, P_{k|k})$, the adequate vector $\overrightarrow{D} = (T_y, T_z, \phi)^T$ made up with the finite translations, T_y, T_z, and rotation, ϕ, which maximizes the variance, $S_{k+1|k}$, of the next predicted measurement pdf. An expression $S_{k+1|k} = F_k(T_y, T_z, \phi)$ of this variance would be very involved, and so would be its closed-form maximization. Hence, an analytic approximation of its gradient around

$\overrightarrow{D_0} = (0,0,0)^T$ is proposed in such way that it points out the direction of its locally steepest ascent.

First, the sigma-points, $\{X_i^-\}$, corresponding to $p(x_k|y_{1:k}) = \mathcal{N}(x_k; \hat{x}_{k|k}, P_{k|k})$, are computed from the posterior mean, $\hat{x}_{k|k}$, and the Cholesky decomposition, $P_{k|k} = L_{k|k}L_{k|k}^T$, of the posterior covariance of the state vector at time k, that is,

$$\{X_i^-\} = \text{Sigma\_points}\left(\hat{x}_{k|k}, L_{k|k}\right) . \tag{2.42}$$

When the binaural head undergoes the rigid motion defined by $\overrightarrow{D} = (T_y, T_z, \phi)^T$ with no dynamic noise, each sigma-point, X_i^+, of the predicted state pdf, $p(x_{k+1}|y_{1:k}) = \mathcal{N}(x_k; \hat{x}_{k+1|k}, P_{k+1|k})$, comes as a function of $\overrightarrow{D}$ and of the corresponding X_i^-, namely,

$$\forall i,\ X_i^+ = \Phi_{X_i^-}(T_y, T_z, \phi) . \tag{2.43}$$

Then, each sigma-point, Y_i^+, of the predicted measurement pdf, $p(y_{k+1}|y_{1:k}) = \mathcal{N}(y_k; \hat{y}_{k+1|k}, S_{k+1|k})$, is obtained from the corresponding sigma-point, X_i^+, as defined in Eq. 2.43 by

$$\forall i,\ Y_i^+ = g\left(\text{atan2}\left(X_i^+(1), X_i^+(2)\right)\right) , \tag{2.44}$$

with $X_i^+(1)$ and $X_i^+(2)$ the entries of X_i^+, $\text{atan2}\left(X_i^+(1), X_i^+(2)\right)$, the azimuth of X_i^+, and $g(\,\cdot\,)$ the Woodworth-Schlosberg formula approximating the interaural time difference (ITD) over a spherical head used for exploration. Finally, the mean, $\hat{y}_{k+1|k}$, and variance, $S_{k+1|k}$, of the predicted measurement pdf are given by the standard unscented transform formulae as follows,

$$\hat{y}_{k+1|k} = \sum_i w_m^i Y_i^+ \tag{2.45}$$

$$S_{k+1|k} = \sum_i w_c^i \left(Y_i^+ - \hat{y}_{k+1|k}\right)^2 . \tag{2.46}$$

To define the gradient with respect to the translation and rotation variables, the first-order Taylor expansions of the functions $\Phi_{X_i^-}(\,\cdot\,)$, $\text{atan2}(\,\cdot\,,\,\cdot\,)$ and $g(\,\cdot\,)$ are composed in the vicinity of $\overrightarrow{D_0}$, considering the infinitesimal translations and rotation vector, $\overrightarrow{du} = (dT_y, dT_z, d\phi)^T$, as

$$\Phi_{X_i^-}(\overrightarrow{D_0} + \overrightarrow{du}) = \Phi_{X_i^-}(\overrightarrow{D_0}) + J\Phi_{X_i^-}(\overrightarrow{D_0}) \cdot \overrightarrow{du} , \tag{2.47}$$

$$\text{atan2}(u, v) = \text{atan2}(u_0, v_0) + \overrightarrow{\nabla}^T \text{atan2}(u_0, v_0) \cdot \begin{pmatrix} u - u_0 \\ v - v_0 \end{pmatrix} , \tag{2.48}$$

$$g(w) = g(w_0) + g'(w_0)(w - w_0) . \tag{2.49}$$

In the above, $\overrightarrow{\nabla}$ stands for the gradient operator, and $J\Phi_{X_i^-}(\overrightarrow{D_0})$ is the Jacobian of $\Phi_{X_i^-}$ evaluated at $\overrightarrow{D_0}$. From the results, $\{Y_i^+(dT_y, dT_z, d\phi)\}$, mean and variance of the predicted measurement pdf follow along Eqs. 2.45, 2.46. This entails the first-order Taylor expansion of $F_k(dT_y, dT_z, d\phi)$, that is,

$$F_k\left(\overrightarrow{D_0} + \overrightarrow{du}\right) = F_k\left(\overrightarrow{D_0}\right) + \overrightarrow{\nabla}^T F_k(\overrightarrow{D_0}) \cdot \overrightarrow{du}\,. \tag{2.50}$$

In the above, $F_k(\overrightarrow{D_0})$ terms the variance of the predicted measurement if no displacement is applied to the head, and $d_k = \overrightarrow{\nabla} F_k(\overrightarrow{D_0})$ is the gradient of F_k evaluated at $\overrightarrow{D_0}$, that is, the local direction of steepest ascent of the variance of the next predicted measurement pdf.

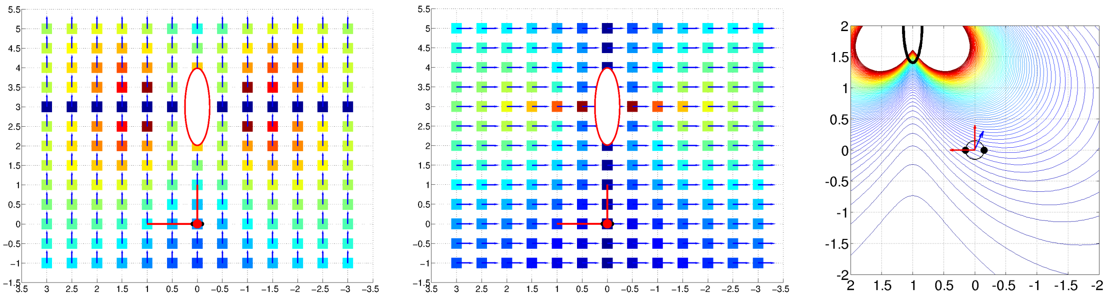
Importantly, function F_{k+1} is not the same as F_k, due to the incorporation of the observation z_{k+1}. In other words, the strategy does not consist in iteratively maximizing the same function over time by the gradient method.

2.8.4 Simulated experiments

Entropy loci of the posterior pdf of the head-to-source situation as a function of head motion

Starting from $p(x_k|y_{1:k}) = \mathcal{N}(x_k; \hat{x}_{k|k}, P_{k|k})$, the entropy, $h(x_{k+1}|y_{1:k+1})$, of the posterior pdf, $p(x_{k+1}|y_{1:k+1})$, were evaluated after applying various sequences of finite translations and rotations to the sensor through u_k. Some results are sketched in Figs. 2.25 a–b. The sensor frame at time k is plotted in red (with $\boldsymbol{e_y}$ and the fovea axis, $\boldsymbol{e_z}$, pointing westwards and northwards, respectively) together with the red 99 %-probability confidence ellipse associated with the sound-source location. In other words, at time k the source was assumed to lie in the vicinity of the sensor fovea, mostly along the boresight axis. The blue arrows depict the fovea direction at time $k+1$ after the movement. $h(x_{k+1}|y_{1:k+1})$ was low (resp. high). This means that the information on the head-to-source situation was high (resp. low) in the warm (resp. cold) areas.

Figure 2.25 c portrays the contours of iso-values of the entropy, $h(y_{k+1}|y_{1:k})$, of the next predicted measurement pdf as a function of the horizontal and vertical positions of a binaural head whose fovea points northwards. The gradient of $h(y_{k+1}|y_{1:k})$ extracted from Eq. 2.50 is also displayed. This case is somewhat similar to Fig. 2.25 a, except that the prior knowledge on the source location, depicted by the black ellipse, was slightly different.



(a) (b) (c)

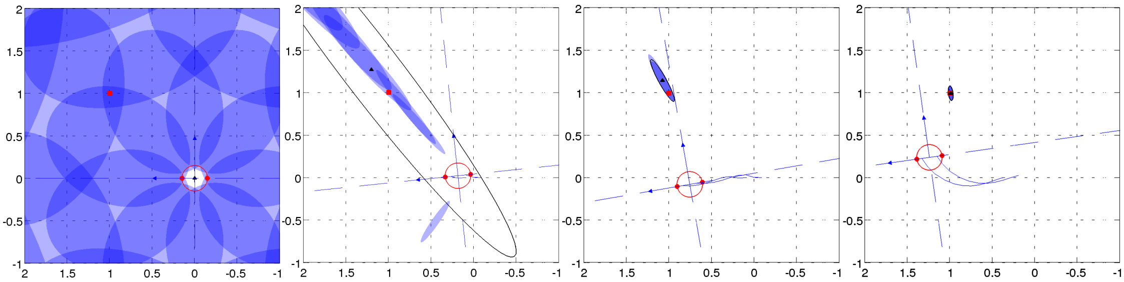
Figure 2.25: Areas of low (warm) and high (cold) entropy for various positions of a binaural head, starting from the initial situation depicted by the red frame (with northwards fovea). The uncertainty of the source situation (before the sensor motion) is described by the red 99 %-probability confidence ellipse. The translation of the sensor is followed by a rotation of, (a), 0 ° or, (b), 90 ° clockwise. (c), iso-contours of the entropy of the next predicted measurement pdf (the higher the entropy, the warmer the color) as a function of the position of a binaural head pointing northwards. The prior knowledge of the source location is depicted by the (partially hidden) black ellipse

Information-based control

MATLAB® simulations were conducted to assess the one-step-ahead gradient-ascent-based control. To this end, GMsrUKF was coupled with the Woodworth-Schlosberg measurement equation for exploration. To avoid simulating binaural signals when the sensor moves, the pseudo likelihood, $p(z_k|\theta_k)$, of the source azimuth mentioned in Sec. 2.8.2 was not extracted from the short-term analysis of the binaural stream, but was replaced by noisy measurements of the genuine source azimuth. The measurement noise was tuned to be significant, in that the spatial uncertainty of the source azimuth corresponding to its $\pm 3\,\sigma$-width amounts to 99%-probability. Confidence intervals range from ±26 ° in the fovea to ±44 ° along the interaural axis – see also Figs. 2.24 c–d.

In Fig. 2.26, the ground-truth position of the sensor (red sphere with ears) is depicted with a blue frame, while the sound-source position is plotted as a red square. The blue ellipses represent the 99%-probability confidence regions associated with the hypotheses of the Gaussian-mixture approximation Eq. 2.33 of the posterior pdf of the head-to-source situation.

Several scenarios were considered for comparing the efficiency of the proposed control strategy – Fig. 2.28. Actually, during the transient mode, the proposed strategy may not be better than purely rotational or circular open-loop motions, because it is based on a single Gaussian pdf, which does not capture all the hypotheses regarding the Gaussian-sum approximation of the genuine posterior pdf of the head-to-source situation. When only one



(a) (b) (c) (d)

Figure 2.26: Audio-motor localization of a sound source by moving the sensor towards the gradient direction of F_k. (a), self-initialization. (b), front-back ambiguity with no motion. (c), information-based motion towards the sound source

hypothesis remains, the decrease becomes more significant.

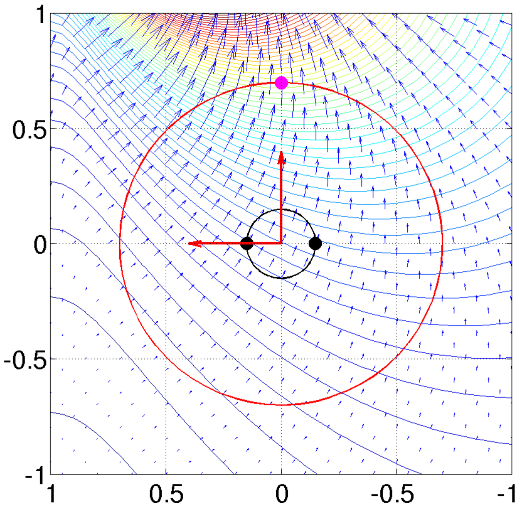


Figure 2.27: Optimum finite translation of binaural head pointing northwards leading to minimum uncertainty in the next head-to-source situation. Iso-contours of the entropy of the next predicted measurement pdf (or, equivalently, of $F_k(\cdot\,,\,\cdot\,,0)$) are plotted, together with their gradient, as a function of the position of a binaural head (in black), whose orientation remains fixed. The red circle delimits the admissible translations. The magenta dot depicts the constrained optimum next to the location of the head

The plotted entropy of the moment-matched approximation of the genuine posterior-state pdf reached a final value that was much lower than the steady-state value of this entropy for other motions. Note that live results from TWO!EARS's robot '*Jido*', being equipped with the motorized Kemar head, are provided in Deliverable D5.2.

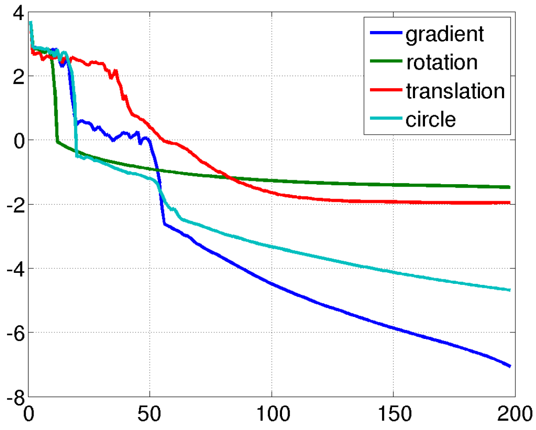


Figure 2.28: Entropy of the moment-matched approximation of the posterior pdf of the head-to-source situation over time for different control strategies. Blue: information based strategy. Green: pure rotation. Red: pure translation. Cyan: circular trajectory around the sound source

2.8.5 Prospects

A forthcoming prospect concerns the implementation of a constrained gradient or Newton algorithm on each F_k to find the admissible optimum *finite* translation and rotation. The constraints of this optimization problem can express the limitations on the velocities of the head – see for instance Fig. 2.27.

Multi-step methods will then be considered. For instance, a criterion based on the expectation of the differential entropy over several steps will be defined to guide the motion – in the vein of [13]. The guidance will thus be viewed as an N–step-optimization problem, where the objective is to find a sequence of robot commands, $u^{\star} = \{u_k, u_{k+1}, ..., u_{k+N}\}$, that improves localization over a sliding time window. The solution may rely on *Partially-Observable Markov Decision Processes*, incorporating reward functions based on information analysis [2] or on other optimum control techniques.

2.9 Controlling parameter settings of the auditory frontend

(The following relates to b1, b2 and b3)

The Auditory Frontend (AFE) of the Two!Ears system supports the modification of many parameters. This makes it, for example, possible to

- 'Sharpen the ears' by controlling the filter bandwidth of the auditory filterbank
- Perform cue selection based on the interaural coherence with the aim of ignoring unreliable interaural time- and level-difference cues
- Activate the Precedence-Effect model in case of strong room reflections

The auditory front-end is implemented as a dynamic chain such that the user, for instance, the blackboard, can either request an additional feature to be computed or modify any of the parameters of an existing feature in between two "blocks" of input ear signal. So far, in practice, the framework architecture allows for new requests. Modifying existing requests is still to be implemented.

2.10 Feedback with regard to the assessment of the Quality of Experience (QoE)

(The following relates to c3)

The following aspects of feedback are relevant for QoE-assessment tasks in the context of Two!Ears.

- Head-orientation for exploratory listening
- Lateral head displacement to identify optimal listening positions
- Specific focusing on auditory features as relevant for quality evaluation task
- Internal reference adaptation (different groups of listeners)

Currently, respective data is missing, but some will hopefully be provided in coming months. To this end, psychoacoustic experiments are being performed. In these experiments, data are collected regarding cognitive tasks such as 'learning internal references', 'weighting different low-level attributes' and 'exploring the sound field'. However, an overall dataset to perform all the steps identified above, will not be available in the short run.

2.11 Auditory features inferred from visual evidence

(The following relates to c6)

In order to simulate cognitive top-down processes, a prototype model was developed that can integrate audio-visual cues and steer the auditory stages based on the visual input. So far, the visual stimuli were generated on the computer, but the plan is to record the visual stimuli later with a stereoscopic camera which is integrated into our binaural manikin. The model estimates left and right corners of a room based on the camera images and uses these measurements to predict acoustical values including room size, angles of incidence, and delay times of the first lateral reflections. The visual model is used in two ways. Firstly, the estimated room coordinates were used to predict the volume of the visual space to calculate the expected values for reverberation time and direct-to-reverberant-energy ratio – thus simulating the results of a previous study by Valente and Braasch [40]. Secondly, the visual model was used as a front end for a novel Precedence-Effect model that suppresses early room reflections by referring to the visual input – which is an example of a top-down process.

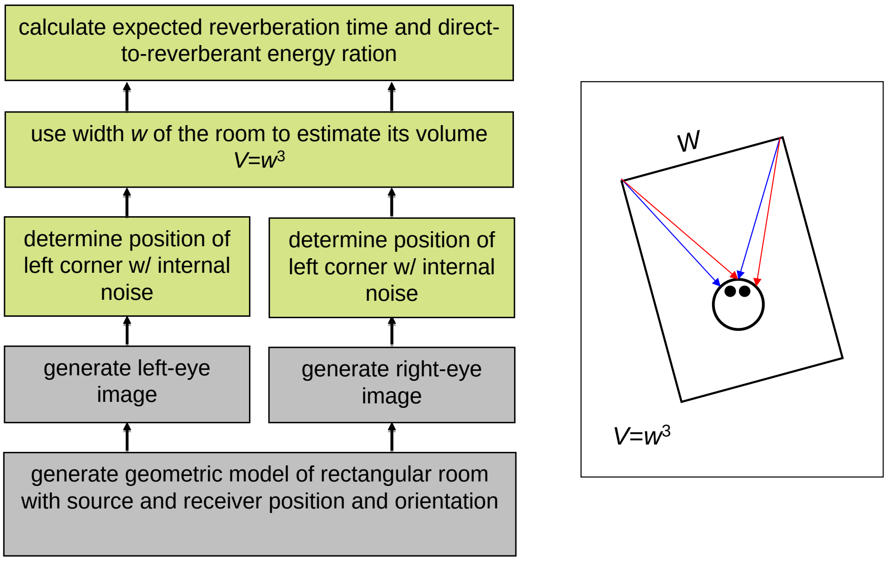

calculate expected reverberation time and direct-to-reverberant energy ration
use width *w* of the room to estimate its volume *V*=*w*³
determine position of left corner w/ internal noise
determine position of left corner w/ internal noise
generate left-eye image
generate right-eye image
generate geometric model of rectangular room with source and receiver position and orientation
W
V=*w*³

Figure 2.29: Architecture of a model to estimate expected reverberation times and early-to-late reverberant-energy ratios from the estimated room volume as were observed in a prior perceptual study [40]

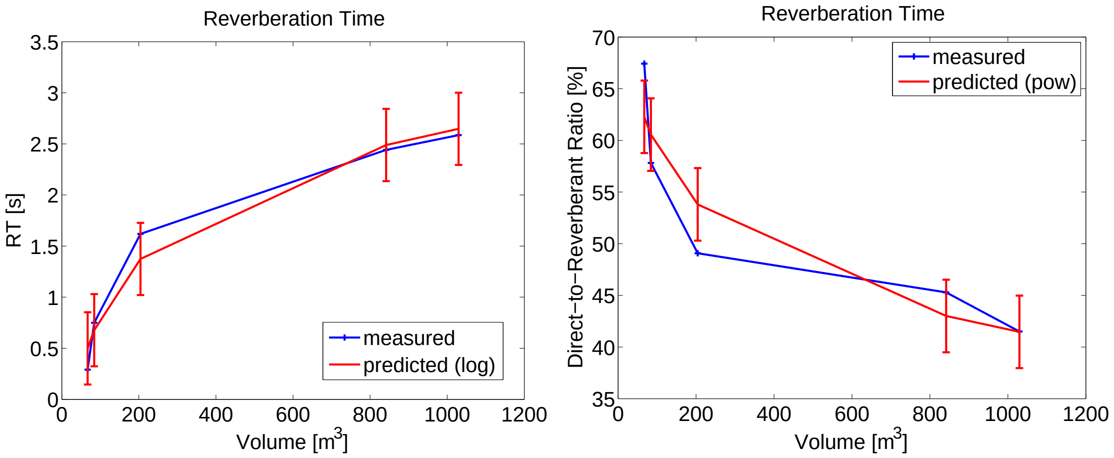


Figure 2.30: Left: Model prediction for the perceived reverberation time (perceptually, the reverberance duration) as a function of estimated room volume (red curve); Right: Model prediction of the early-to-late reverberant-energy ratio (perceptually, the loudness difference of direct and reverberant sounds) as a function of the estimated room volume (red curve). In both cases, the blue curves show the underlying perceptual data representing the expectation of human listeners

2.11.1 Predicting visual expectation of reverberation time and direct-to-reverberant energy ratio

Figure 2.29 shows the visual-expectation model. At first the visual cues were generated. Then a visual model of a rectangular room was computed, based on a given length, width and height of the room – see bottom left. Next, the left- and right-eye signals are computed from the room geometry and the viewer's position and orientation. For this purpose, the room edges were drawn in MATLAB®, using a line model, and stored as bitmap images. The bitmaps were then analyzed by the visual model. First, the left vertical corner edge was settled for the left eye to determine the azimuth angle of this edge in the head-related coordinate system – see right side of Fig. 2.29. The same procedure was applied to determine angles for the right-edge as well as the left- and right-corner-edge angles, measured from the right-eye perspective. For all four angle measurements, internal noise was introduced at this point to limit the accuracy of the angle estimation. Based on the four angles, the distances of both edges could be computed to determine the actual width of the room. From there the volume of the room was simply estimated as the width cubed. Although this is very crude approximation, one has to consider that, (i), humans usually estimate the size of rooms without inspecting all room edges and, (ii), the participants in the Valente-&-Braasch study [40] had to estimate the room sizes from photos without complete information about the room geometry.

In the next step, a straight line was fitted through the logarithmized expected-reverberation-time values as a function of room volume using linear regression and the perceptual data from [40]. The left graph of Fig. 2.30 shows the perceptual results (blue) and the simulated data (red). The latter were derived from the visually-estimated room volume and the measured regression line. The right panel of Fig. 2.30 shows the measured and simulated data for the expected direct-to-reverberant-energy ratios. Also in this case, linear regression of the logarithmized physical data (i.e., the direct-to-reverberant energy ratios) was applied to simulate the human perceptual data (i.e., the loudness differences of direct and reverberant sounds).

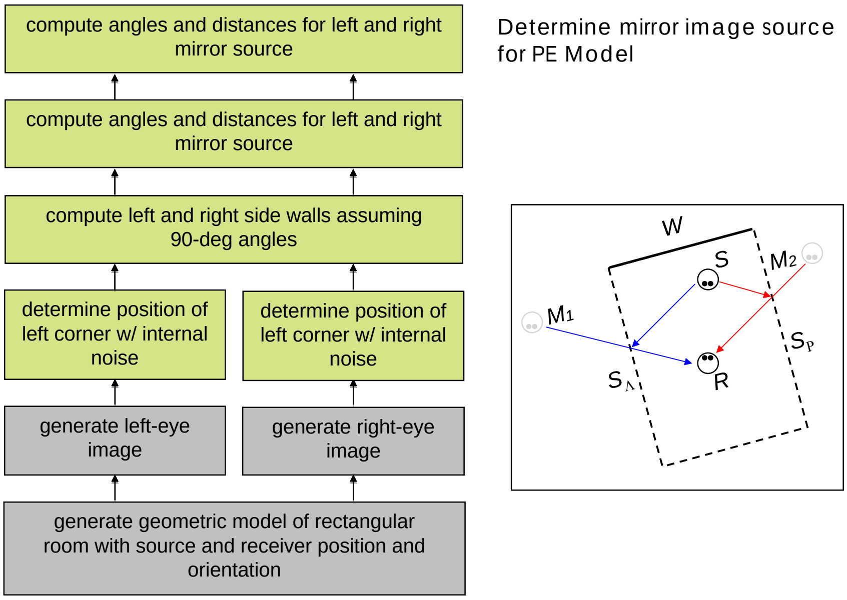

compute angles and distances for left and right mirror source
compute angles and distances for left and right mirror source
compute left and right side walls assuming 90-deg angles
determine position of left corner w/ internal noise
determine position of left corner w/ internal noise
generate left-eye image
generate right-eye image
generate geometric model of rectangular room with source and receiver position and orientation
Determine mirror image source for PE Model
W
S
M2
M1
SP
SA
R

Figure 2.31: Architecture of a model to estimate the angles of incidence and arrival times of the two first-order lateral reflections from visual cues

2.11.2 Visual top-down mechanism for lag suppression in the Precedence Effect

These visually-extracted parameters were then used by a binaural model to suppress the early acoustic reflections as to simulate the visual build-up of the Precedence Effect. For this purpose, the visual-room-perception model was extended to predict the location of the left and right side walls – as shown in Fig. 2.31. The model continues where the previous model left off with the estimation of the locations of the left and right side walls – see right

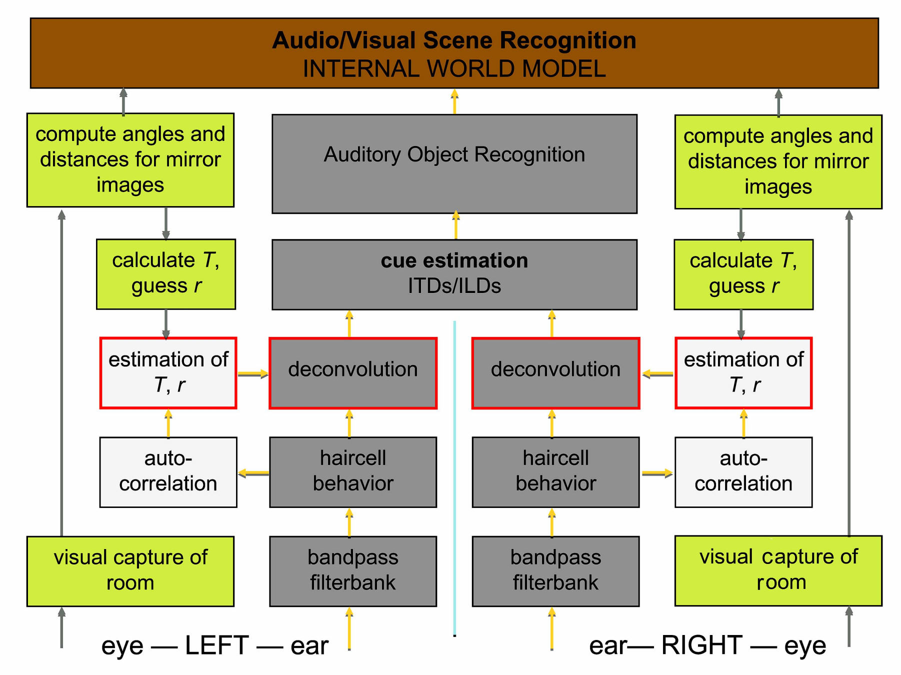


Figure 2.32: Architecture of a Precedence-Effect (PE) localization model to demonstrate the PE build-up effect from visual cues using a combined auditory-visual model. The model calculates the delay, T, of the early lateral reflections from the visually-captured room geometry for the left- and right-ear channels. It then estimates the reflection coefficients, r, to judge the amplitudes of the reflection to partially remove the reflections from the signal using a filter according to [5]

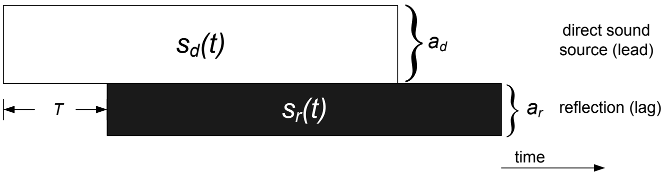


Figure 2.33: Time course of a single-channel lead/lag pair that consists of a direct sound source, $s_d(t)$, with amplitude a_d and one reflection, $s_r(t)$, with amplitude a_r. The reflection is delayed by delay time T

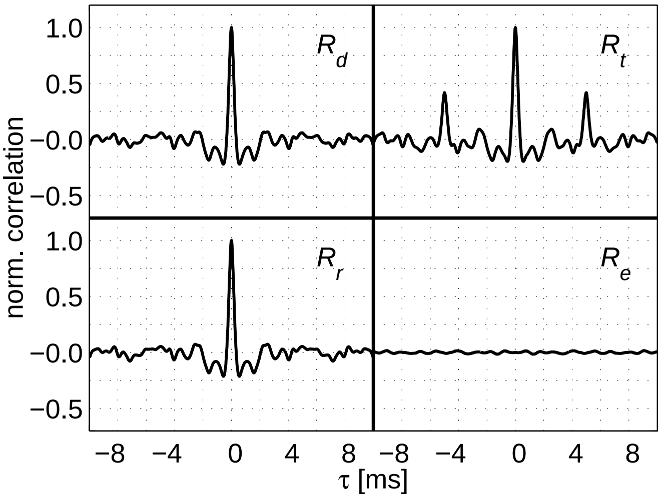


Figure 2.34: Autocorrelation functions for various broadband-signal configurations, namely,: direct signal only (top-left panel, R_d), total signal, direct sound with reflection (top-right panel, R_t), reconstructed direct sound consisting of total signal with eliminated reflection (bottom-left panel, R_r), and the error between reconstructed sound and its original (bottom-right panel, $R_e = R_r - R_d$). If applicable, the LLAR was set to 0.5, while the delay between lead and lag was set to 5 ms

side of Fig. 2.31. Here, the height and length of the walls are not important, but only the angles and distances of the walls respective to the receiver position. The angles of the walls were calculated from the angle of the front wall based on the two previously measured corner angles. It was assumed that the side angles are perpendicular to the front angle. The outcome of the model was used to determine the distance and incidental angles of the two early side reflections using the mirror image model and a given source position, S. The coordinates for the side reflections (distance and angle of incidence) were then passed on to the auditory module of the model.

Figure 2.32 shows how the auditory and visual modules interact. An existing Precedence-Effect model from 2013 [5], which will be described in the next section, served as the basis for the auditory module.

2.11.3 Basic Concepts

Identification of ISI and lag amplitude

Although localization dominance is usually attributed to binaural effects, it is easier to first outline the model algorithm by using a monophonic example of a direct signal, $s_d(t)$, and a single reflection, $s_r(t)$. Since a reflection is a delayed copy of a direct signal, one can write

$$s_r(t) = r \cdot s_d(t - T)\,, \tag{2.51}$$

with the delay time, T, and the Lead/Lag Amplitude Ratio (LLAR), r. The latter can be treated as a frequency-independent, phaseshift-less-reflection coefficient, given that the decrease in sound pressure with distance of lead and lag can be neglected. For a passive reflection, one typically finds $r \leq 1$. At least this is the case when the direct sound source and the reflection are captured with a (hypothetical) omnidirectional receiver. In the psychoacoustics literature, the delay time, T, is often referred to as the inter-stimulus interval (ISI). The total signal, $s_t(t)$, which consists of the direct sound, $s_d(t)$, and its reflection, $s_r(t)$, can mathematically be described as follows – see also Fig. 2.33,

$$s_t(t) = s_d(t) + s_r(t) = s_d(t) + r \cdot s_d(t - T). \tag{2.52}$$

In the next step, the autocorrelation function of the total signal was used to extract information about both the delay time, T, and the LLAR, r, as follows,

$$\begin{aligned} R_{s_t} &= \int\limits_{-\infty}^{\infty} s_t(t) \cdot s_t(t - \tau)\, dt \\ &= R_{s_d} + R_{s_r} + R_{s_d s_r} + R_{s_r s_d}\,. \end{aligned} \tag{2.53}$$

Aside from the two cross-correlation terms, there are now two autocorrelation terms available, one for the direct sound, R_{s_d}, and one for the reflection, R_{s_r}. In case that the direct sound is aperiodic, both functions should only have one peak, located at $\tau = 0$.

The top-left panel of Fig. 2.34 shows the autocorrelation peak for a broadband-noise signal (direct sound only). The lead/lag condition is shown in the top-right panel of Fig. 2.34. Since the direct sound and its reflection are highly correlated with each other, two cross-correlation terms, $R_{s_d s_r}$ and $R_{s_r s_d}$, were received. The first one has its maximum at $\tau = -T$, the second one at $\tau = T$. Hence, for aperiodic signals the following holds,

$$R_{s_t} = \left\{ \begin{array}{rcl} r s_d^2 & : & \tau = -T \\ \left(1 + r^2\right) s_d^2 & : & \tau = 0 \\ r s_d^2 & : & \tau = +T \end{array} \right\} . \tag{2.54}$$

The delay time between direct sound and reflection could easily be estimated by determining the position of one of the two side peaks. The next task was to determine the LLAR, r, from the ratio, γ, between one of the autocorrelation side peaks and the main autocorrelation peak, namely,

$$\gamma = \frac{R_{s_r s_d}}{R_{s_d} + R_{s_r}} = \frac{r s_d^2}{(1+r^2)\, s_d^2} = \frac{r}{(1+r^2)\ .} \tag{2.55}$$

By completing the square, Eq. 2.55 can be resolved for r as follows,

$$r = \pm\sqrt{\frac{1}{4\gamma^2} - 1} + \frac{1}{2\gamma}\,. \tag{2.56}$$

The ambiguities will be dealt with later in Sect. 2.11.3.

Lag removal through deconvolution

Now that the delay between lead and lag, T, and the LLAR, r, were known, a simple filter could be designed, which eliminated the lag from the total signal. Interestingly, this solution coincides with the impulse response of a cylindrical pipe resonator. The

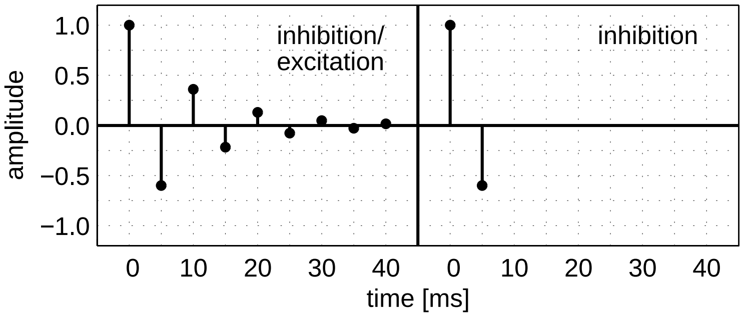


Figure 2.35: Impulse response of the novel lag-suppression filter for eight iterations (left panel), as it was applied to remove the lag in Fig. 2.34. The right panel shows the equivalent implementation for a simple approach with ipsilateral inhibition and no excitatory elements

deconvolution filter, h_d, converged fairly fast and only a few iterations, N, were needed, that is,

$$h_d = \sum_{n=0}^{N} (-r)^n \, \delta(T \cdot n) \, . \tag{2.57}$$

Of course, in the ideal case, N approaches ∞. The filter's mode of operation is in fact fairly intuitive. The main peak of the filter lets the complete signal pass, while the first negative peak is adjusted to eliminate the lag by subtracting a delayed copy of the signal. However, one has to keep in mind that also the reflection will be processed through the filter and, thus, the second, negative delta peak will evoke a further signal component, which is delayed by $2T$ compared to the direct signal. This newly generated component has to be compensated by a third, positive peak of the filter. A number of iterations were necessary to reduce the artifacts that result from the previous peaks. It is obvious that r cannot approximate one, as otherwise the filter would not converge. For LLARs close to one, it is thus advantageous to limit r in Eq. 2.57.

In other current models, the delay between the first, positive and second, negative peak was typically set as constant and not optimized for different stimuli – see, Fig. 2.35. Also, the magnitude of the negative peak was set globally. However, in the case of the autocorrelation-based approach used here, the parameters of the filter were optimized for the amplitude ratio between lead and lag and the delay time between both. Further, the system response was no longer a plain inhibitory mechanism, but rather one that included both inhibitory and excitatory elements.

The bottom-left panel of Fig. 2.34 shows the autocorrelation result for a deconvolved signal. In this graph, the side peaks of the autocorrelation function as visible in the top-right panel disappeared fully, and the function is very similar to the autocorrelation function of the lead only – as is plotted in the top-left panel of the same figure.

ITD-based signals

Thus far, a binaural mechanism has not yet been specified to demonstrate localization dominance by any means. This is, because the focus was put on a monaural algorithm to filter out the lag of the total signal. To demonstrate this effect, now the algorithm for a simple cross-correlation model was applied to determine the interaural cross-correlation (ICC) function, namely,

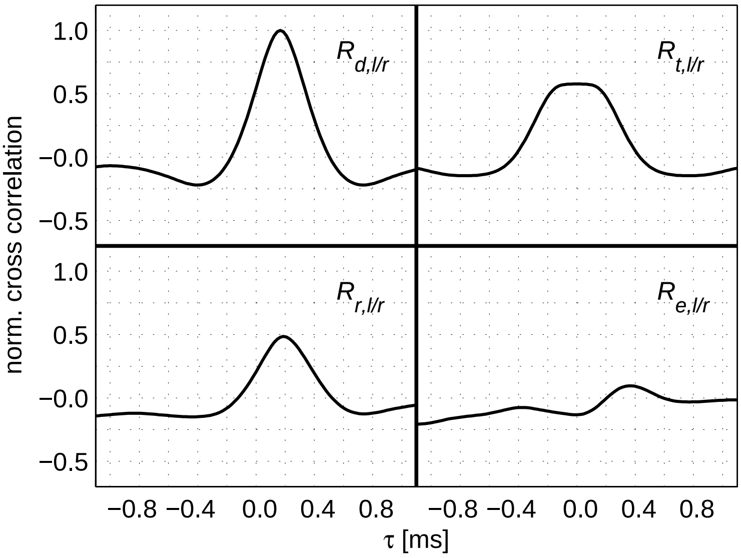


Figure 2.36: Interaural cross-correlation functions (ICC) for a binaural lead/lag pair based on ITD cues. The top-left panel shows the ICC for the direct sound only, while the top-right panel depicts the situation for an additional lag sound. In the bottom-left panel, the lag has been removed from both channels using lag-suppression filters according to Eq. 2.57 before the ICC was calculated. The bottom-right panel shows the difference between the original ICC for the direct sound and the reconstructed one after lag removal

$$R_{s_l s_r} = \int_{-\infty}^{\infty} s_l(t) \cdot s_r(t-\tau)\,dt\,, \tag{2.58}$$

with the left- and right-ear signals, s_l and s_r.

A typical binaural lead/lag pair was created by applying an interaural time difference to the lead signal and processing the lag with another ITD of the same magnitude but of opposite sign. As the ISI is commonly defined as the delay time between lead and lag – without considering spatial processing by applying ITDs – the actual delay times between lead and lag at both ear signals do not have to match the ISI completely. Usually, ITDs are applied in such a way that the signal is preceded by half the ITD value in one channel and delayed by the same value in the opposite channel. Accordingly, T and r for each channel were estimated individually. Here, the deconvolution filters for both channels were determined and both channels were then convolved separately. Consequently, the deconvolved binaural-signal pair was processed with the localization model. Alternatively, separate filters could have been applied to the left and right ear signal before finally calculating ICC.

Figure 2.36 depicts the calculated cross-correlation functions. The top-left panel shows the ICC for a single sound with an ITD of 330 μs (100-ms broadband-noise burst with a frequency range of 200–1000 Hz). The position of the cross-correlation peak clearly indicates the ITD of the stimulus. In the top-right panel, the same sound is accompanied by a reflection (r=1, -330-μs ITD, 5-ms ISI). A single peak was still observed, being located in between the location of both sounds. The bottom-left panel depicts the stimulus after the lag was removed with the lag-suppression filter in both ear signals. Now the location of the ICC peak corresponded again to the ITD of the lag and, thus, the algorithm was demonstrating localization dominance. The bottom-right panel of Fig. 2.36 presents the negligible error between the original ICC function for the lead and the reconstructed function after lag removal, that is, $R_e = R_d - R_r$.

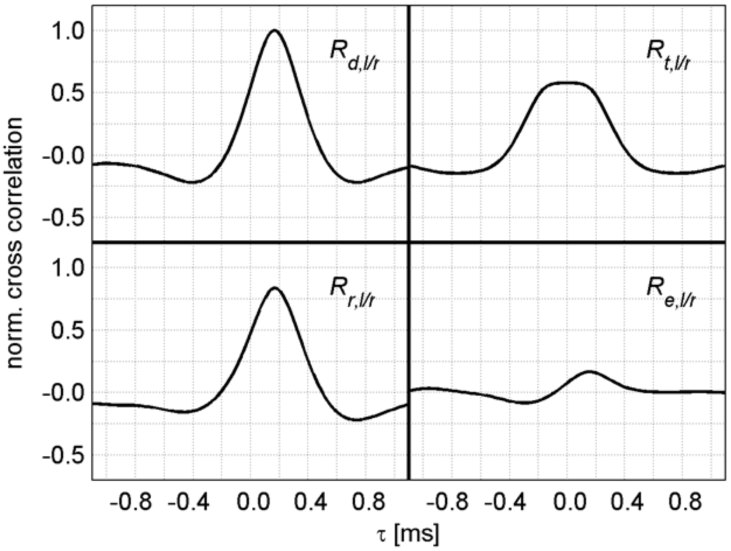


Figure 2.37: Result of the Precedence-Effect model with optimal visual input

Currently, the success of the algorithm depends too strongly on the accurate estimation of the arrival times of the two lateral reflections. For accurate estimation the error needs to be within a few milliseconds – which is probably an unrealistic assumption of the brain's performance.

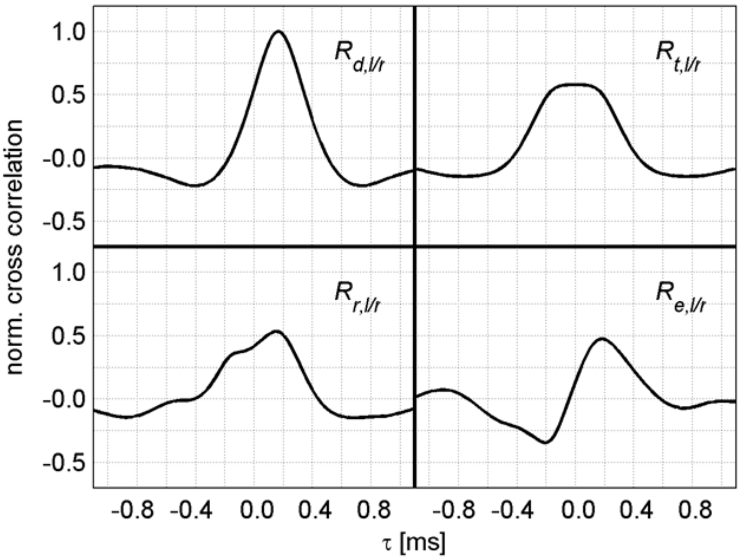


Figure 2.38: Precedence-Effect-model results with 2 %-delay error and 50 %-amplitude error

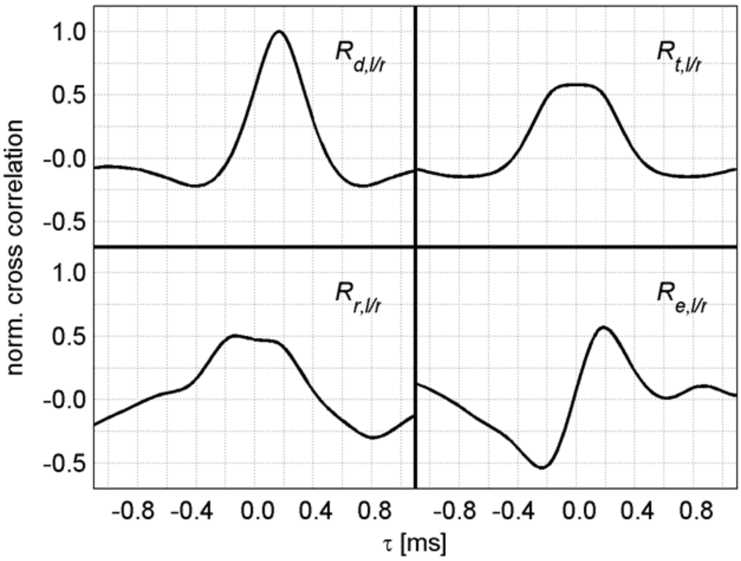


Figure 2.39: Result for Precedence-Effect model with a 20 %-delay error and a 50 %- amplitude error

Bibliography

[1] Aaronson, N. and Hartmann, W. (**2014**), “Testing, correcting, and extending the Woodworth model for interaural time difference,” *The Journal of the Acoustical Society of America* **135**, pp. 817–823. (Cited on page 59)

[2] Araya-López, M., Buffet, O., Thomas, V., and Charpillet, F. (**2010**), “A POMDP Extension with Belief-dependent Rewards,” in *Advances in Neural Information Processing Systems 23*, Curran Associates, Inc., pp. 64–72. (Cited on page 67)

[3] Blauert, J., Kolossa, D., and Danés, P. (**2014**), “Feedback loops in engineering models of binaural listening.” in *Proc. Meetings Acoust., POMA 21*, paper 1pPP11. (Cited on page 57)

[4] Bourgault, F., Makarenko, A., Williams, S., Grocholsky, B., and Durrant-Whyte, H. (**2002**), “Information based adaptive robotic exploration,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, (IROS'2002)*, Lausanne, Switzerland. (Cited on page 57)

[5] Braasch, J. (**2013**), “A precedence effect model to simulate localization dominance using an adaptive, stimulus parameter-based inhibition process,” *J. Acoust. Soc. Am.* **134**(1), pp. 420–435. (Cited on pages 47, 48, 72, and 73)

[6] Brown, G. and Cooke, M. (**1994**), “Computational auditory scene analysis,” *Comput. Speech. Lang.* **8**, pp. 297–336. (Cited on page 39)

[7] Bustamante, G., Portello, A., and Danès, P. (**2015**), “A Three-Stage Framework to Active Source Localization from a Binaural Head,” in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'2015)*, Brisbane, Australia. (Cited on page 56)

[8] Clark, N. R., Brown, G. J., Jürgens, T., and Meddis, R. (**2012**), “A frequency-selective feedback model of auditory efferent suppression and its implications for the recognition of speech in noise.” *The Journal of the Acoustical Society of America* **132**(3), pp. 1535–41, URL `http://www.ncbi.nlm.nih.gov/pubmed/22978882`. (Cited on page 45)

[9] Cohen-Lhyver, B., Argentieri, S., and Gas, B. (**2015**), “Modulating the Auditory Turn-to Reflex on the Basis of Multimodal Feedback Loops : the Dynamic Weighting Model,” in *IEEE-ROBIO*, Zhuhai, China. (Cited on pages 28 and 34)

[10] Cooke, M., Barker, J., Cunningham, S., and Shao, X. (**2006**), "An audio-visual corpus for speech perception and automatic speech recognition," *J. Acoust. Soc. Am.* **120**, pp. 2421–2424. (Cited on page 41)

[11] Cover, T. and Thomas, J. (**1991**), *Elements of Information Theory*, Wiley. (Cited on pages 57, 60, and 61)

[12] Denzler, J. and Brown, C. (**2002**), "Information Theoretic Sensor Data Selection for Active Object Recognition and State Estimation," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **24**(2), pp. 145–157. (Cited on page 57)

[13] Deutsch, B., Zobel, M., Denzler, J., and Niemann, H. (**2004**), "Multi-step Entropy Based Sensor Control for Visual Object Tracking," *Pattern Recognition* **3175**, pp. 359–366. (Cited on page 67)

[14] Forster, C., Pizzoli, M., and Scaramuzza, D. (**2014**), "Appearance-based Active, Monocular, Dense Reconstruction for Micro Aerial Vehicles," in *Proceedings of Robotics: Science and Systems*, Berkeley, USA. (Cited on page 57)

[15] Gaese, B. H. and Wagner, H. (**2002**), "Precognitive and cognitive elements in sound localization," *Zoology* **105**, pp. 329–339. (Cited on page 38)

[16] González, J. A., Peinado, A. M., Gómez, A. M., and Ma, N. (**2012**), "Log-spectral feature reconstruction based on an occlusion model for noise robust speech recognition," in *Proc. Interspeech*, pp. 2630–2633. (Cited on page 40)

[17] Grocholsky, B., Makarenko, A., and Durrant-Whyte, H. (**2003**), "Information-theoretic coordinated control of multiple sensor platforms," in *IEEE Int. Conf. on Robotics and Automation, (ICRA'03)*, Taipei, Taiwan. (Cited on page 57)

[18] Julian, B. (**2013**), "Mutual Information-based Gradient-ascent Control for Distributed Robotics," Ph.D. thesis, Massachusetts Institute of Technology. (Cited on page 57)

[19] Kumon, M., Fukushima, K., Kunimatsu, S., and Ishitobi, M. (**2010**), "Motion planning based on simultaneous perturbation stochastic approximation for mobile auditory robots," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'2010)*, Taipei, Taiwan. (Cited on page 57)

[20] Liberman, M. C. (**1988**), "Response properties of cochlear efferent neurons: monaural vs. binaural stimulation and the effects of noise," *Journal of Neurophysiology* **60**(5), pp. 1779–1798, URL `http://jn.physiology.org/content/60/5/1779`. (Cited on page 45)

[21] Ma, N., Brown, G. J., and Gonzalez, J. A. (**2015**), "Exploiting top-down source models to improve binaural localization of multiple sources in reverberant environments," in

Proceedings of Interspeech, Brisbane. (Cited on page 40)

[22] Manyika, J. (**1993**), "An Information-Theoretic Approach to Data Fusion and Sensor Management," Ph.D. thesis, University of Oxford. (Cited on page 60)

[23] Martinson, E., Apker, T., and Bugajska, M. (**2011**), "Optimizing a reconfigurable robotic microphone array," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'2011)*, San Francisco, California. (Cited on page 57)

[24] May, T., Ma, N., and Brown, G. J. (**2015**), "Robust localisation of multiple speakers exploiting head movements and multi-conditional training of binaural cues," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.* (Cited on page 41)

[25] May, T., van de Par, S., and Kohlrausch, A. (**2013**), "Binaural Localization and Detection of Speakers in Complex Acoustic Scenes," in *The technology of binaural listening*, edited by J. Blauert, Springer, Berlin–Heidelberg–New York NY, chap. 15, pp. 397–425. (Cited on page 41)

[26] Nakadai, K., Lourens, T., Okuno, H., and Kitano, H. (**2000**), "Active Audition for Humanoid," in *Nat. Conf. on Artificial Intelligence (AAAI'2000)*, Austin, TX. (Cited on page 56)

[27] N.N. (**2015**), "Signal Metrics," URL `http://www.dsprelated.com/freebooks/mdft/Signal_Metrics.html`. (Cited on page 12)

[28] OGRE (**2014**), "OGRE - Open Source 3D Graphics Engine," URL `http://www.ogre3d.org/`. (Cited on page 10)

[29] Poggio, T. and Bizzi, E. (**2004**), "Generalization in vision and motor control," *Nature* **431**(7010), pp. 768–774. (Cited on page 22)

[30] Portello, A., Bustamante, G., Danès, P., and Mifsud, A. (**2014**), "Localization of Multiple Sources from a Binaural Head in a Known Noisy Environment," in *IEEE/RSJ Int. Conf. on Intell. Robots and Systems (IROS'2014)*, Chicago, IL. (Cited on page 57)

[31] Portello, A., Bustamante, G., Danès, P., Piat, J., and Manhès, J. (**2014**), "Active Localization of an Intermittent Sound Source from a Moving Binaural Sensor," in *Forum Acustium (FA'2014)*, Krakow, Poland. (Cited on pages 57 and 58)

[32] Portello, A., Danès, P., and Argentieri, S. (**2012**), "Active Binaural Localization of Intermittent Moving Sources in the Presence of False Measurements," in *IEEE/RSJ Int. Conf. on Intell. Robots and Systems (IROS'2012)*. (Cited on page 57)

[33] Portello, A., Danès, P., Argentieri, S., and Pledel, S. (**2013**), "HRTF-Based Source Azimuth Estimation and Activity Detection from a Binaural Sensor," in

IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'2013), Tokyo, Japan. (Cited on pages 56 and 58)

[34] Rennie, S. J., Hershey, J. R., and Olsen, P. A. (**2010**), "Single-channel multitalker speech recognition," *IEEE Signal Process. Mag.* **27**, pp. 66–80. (Cited on page 40)

[35] Sommerlade, E. and Reid, I. (**2008**), "Information-theoretic active scene exploration," in *IEEE Conf. on Computer Vision and Pattern Recognition, (CVPR'2008)*, Anchorage, Alaska. (Cited on page 57)

[36] Spence, G. C. and Driver, J. (**1994**), "Covert spatial orienting in audition: exogenous and endogenous mechanisms," *Journal of Experimental Psychology* **20**, pp. 555–574. (Cited on page 38)

[37] Stern, R. M. (**1988**), "Lateralization of complex binaural stimuli: A weighted-image model," *The Journal of the Acoustical Society of America* **84**(1), pp. 156–165, URL `http://scitation.aip.org/content/asa/journal/jasa/84/1/10.1121/1.396982`. (Cited on page 48)

[38] Thrun, S., Burgard, W., and Fox, D. (**2005**), *Probabilistic Robotics*, The MIT Press. (Cited on page 57)

[39] University of London – School of Electronic Engineering and Computer Science (**2015**), "IEEE AASP Challenge," URL `http://c4dm.eecs.qmul.ac.uk/sceneseventschallenge/`. (Cited on page 7)

[40] Valente, D. L. and Braasch, J. (**2008**), "Subjective Expectation Adjustments of Early-to-Late Reverberant Energy Ratio and Reverberation Time to Match Visual Environmental Cues of a Musical Performance," *Acta Acustica United with Acustica* **94**, pp. 840–855. (Cited on pages 69, 70, and 71)

[41] Varga, A. and Moore, R. (**1990**), "Hidden Markov model decomposition of speech and noise," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, pp. 845–848. (Cited on page 40)

[42] Vincent, E., Sini, A., and Charpillet, F. (**2015**), "Audio source localization by optimal control of a mobile robot," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'2015)*, Brisbane, Australia. (Cited on page 57)

[43] Walther, T. and Cohen-L'hyver, B. (**2014**), "Multimodal feedback in auditory-based active scene exploration," in *Proc. Forum Acusticum*, Kraków, Poland. (Cited on pages 6, 14, and 28)

[44] Wang, D. L. and Brown, G. J. (Eds.) (**2006**), *Computational Auditory Scene Analysis: Principles, Algorithms and Applications*, Wiley/IEEE Press. (Cited on page 38)

[45] Winkowski, D. E. and Knudsen, E. I. (**2006**), “Top-down gain control of the auditory space map by gaze control circuitry in the barn owl.” *Nature* **439**(7074), pp. 336–339. (Cited on page 38)

[46] Woodruff, J. and Wang, D. L. (**2012**), “Binaural localization of multiple sources in reverberant and noisy environments,” *IEEE Trans. Audio, Speech, Lang. Process.* **20**(5), pp. 1503–1512. (Cited on page 41)