Real-time sound localization with a binaural head-system using a biologically-inspired cue-triple mapping

Tobias Rodemann, Martin Heckmann, Björn Schölling, Frank Joublin, Christian Goerick

2006

Accepted at IROS 2006, 9.–15. October 2006 in Beijing

Real-time sound localization with a binaural head-system using a biologically-inspired cue-triple mapping

Tobias Rodemann, Martin Heckmann, Frank Joublin, Christian Goerick
Honda Research Institute Europe GmbH
Carl-Legien-Str. 30
D-63073 Offenbach/Main, Germany
{Tobias.Rodemann,Martin.Heckmann,Frank.Joublin,Christian.Goerick}@honda-ri.de

Björn Schölling
Institut für Automatisierungstechnik
Technische Universität Darmstadt
D-64283 Darmstadt, Germany
bjoern.schoelling@rtr.tu-darmstadt.de

***Abstract*— We present a sound localization system that operates in real-time, calculates three binaural cues (IED, IID, and ITD) and integrates them in a biologically inspired fashion to a combined localization estimation. Position information is furthermore integrated over frequency channels and time. The localization system controls a head motor to fovealize on and track the dominant sound source. Due to an integrated noise-reduction module the system shows robust localization capabilities even in noisy conditions. Real-time performance is gained by multi-threaded parallel operation across different machines using a timestamp-based synchronization scheme to compensate for processing delays.**

I. Introduction

Sound localization in a real-world environment (see e.g. [1]–[4]) is a hard problem, requiring an integration of different modules into a system that runs in real-time. A number of approaches (e.g. [2], [5]) use special sensing or computing hardware to solve this problem. In contrast to this we present an architecture that uses a humanoid head with just two microphones and reaches real-time capabilities using standard computing hardware. Robustness regarding noise and echoes is achieved by using measurement window selection (detailed in [6]), a static noise reduction system and integration over frequency channels, localization cues and time.

There are three cues for the relative position of a sound source: the Interaural Time Difference (**ITD**), the Interaural Intensity Difference (**IID**) and finally the Interaural Envelope Difference (**IED**). Each of these cues has its drawbacks: ITD only works for the lower frequency range, becoming ambiguous beyond a critical frequency of around 1kHz (depending on head dimensions), IID is strong only for the higher frequencies and very much dependent on the hardware characteristics of head and microphones, and IED is generally considered to be quite unreliable on its own [7]. In addition both echoes and additional noise sources degrade system operation. Sound localization therefore requires an integration of information from different sources at the correct time. As a target application we set a scenario with a varying number of auditory sources (e.g. humans) in a normal (noisy and echoic) room. People are supposed to address the system through calls, claps, whispers or any other sound. The system should turn immediately to the position of the currently strongest sound source, while ignoring static noise sources like air condition or fan noise. The design of our system was constrained by the need to add more functionality in the future, therefore requiring a flexible software architecture and general-purpose hardware. While [6] details the cue computation and window selection algorithm used in our system, this paper focuses on three different aspects: integrating localization information, stationary noise reduction, and the software skeleton around which our system is built.

A. Related Approaches

Robot sound localization has been presented before, however with a different hardware and software structure and a different focus. Many authors used microphone arrays [5], [8] to get a satisfactory performance under real-world conditions. Our system uses only two microphones mounted on a humanoid head. We also use conventional computing hardware (single CPU systems or SMP machines) instead of dedicated hardware. Despite of this we are capable of using the computationally more expensive Gammatone filters [9] which are considered to be a good approximation of processing in the human cochlea and provide a high resolution in time and frequency. We are using zero-crossings [10] to measure ITD and IED. Also inspired from the biological example is the integration of cues in a neuron-like manner. Another biologically-inspired approach has been presented in [11], which focuses on a probabilistic estimation of sound source position, but not on the capability to work in a real-world scenario. Therefore, our system is special in the sense that it provides a biologically-inspired binaural sound localization with the necessary robustness to operate in real-world environments.

II. System Architecture

The system consists of different processing modules, which can be grouped into sound recording, preprocessing, cue extraction, cue mapping, integration and neck control elements. For a view of the complete graph see Fig. 1. In addition, there are modules for synchronization, latency compensation and downsampling (note that for reasons of clarity synchronization modules are not shown in the graph).

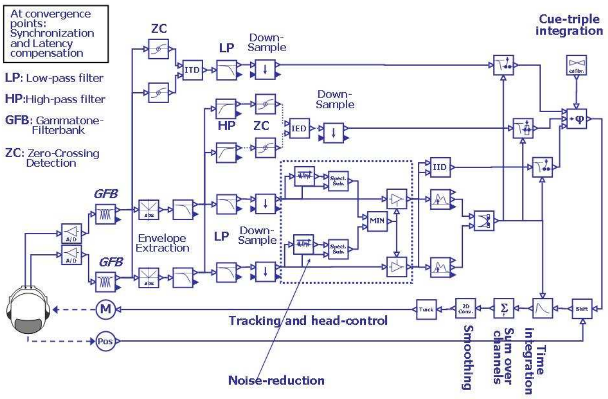

Fig. 1. The complete system graph. For a description see text.

A. Preprocessing

The system records sound data from two microphones mounted on a humanoid head, and then uses a Gammatone filterbank (GFB) [9], [12], [13] with Equivalent-Rectangular Bandwidth (ERB) to get frequency-specific signal responses $g(c,k)$ (channel c, time index k). In our experiments we used between 30 and 180 different frequency channels in the range of 100 Hz to 10 kHz and a sampling rate of 24 kHz. We then compute the signal's envelope $e(c,k)$ through rectification and frequency-specific low pass filtering. We apply a high-pass filtering with a cut-off frequency of 500 Hz on the envelope signal. The resulting signal will be denoted $h(c,k)$. Based on the envelope signal we also compute $l(c,k)$ using a low-pass filter with a cut-off frequency of 40 Hz in order to remove pitch-based amplitude modulations of unresolved harmonics. The noise reduction operates on the low-pass filtered signal $l(c,k)$, producing the noise-reduced signal $s(c,k)$.

B. Localization Cues

Both ITD and IED are based on comparing consecutive zero-crossings from left and right microphones. The comparison is done for every zero-crossing point of one side with the previous and the next zero-crossing on the contra-lateral side. IED is based on zero-crossings taken from the high-pass signal $h(c,k)$, while ITD uses Gammatone filterbank output $g(c,k)$. IID computation is based on a comparison of left and right noise free signals ($n^l(c,k)$ and $n^r(c,k)$):

$$IID(c,k) = \frac{s^l(c,k) - s^r(c,k)}{\max(s^l(c,k), s^r(c,k))} \quad . \tag{1}$$

Cues are computed continuously but measured only at certain times, where echoes have a limited effect. How these measurement windows are computed is described in detail in [6]. The basic approach is a maximum search near signal onsets with an inhibition of trailing maxima. This approach is inspired by the precedence-effect in hearing psychology.

C. Cue Mapping

After computing the three interaural cues we integrate them in a biologically-inspired manner and map them as a cue-triple to different positions along the horizontal (azimuth) axis. For every position i and frequency channel c we define a node with a receptive field $RF_{i,c} =$ ($\text{IED}_{i,c}$, $\text{IID}_{i,c}$, $\text{ITD}_{i,c}$, $\sigma^{IED}_{i,c}$, $\sigma^{IID}_{i,c}$, $\sigma^{ITD}_{i,c}$, $w^{IED}_{i,c}$, $w^{IID}_{i,c}$, $w^{ITD}_{i,c}$) within cue space. Receptive field center, width, and confidence, respectively, are defined using the calibration procedure described below. For a cue-triple (IED,IID,ITD) measured at time index k we compute the response $M_c(i,k)$ of every node i by calculating the distance of the cue-triple to the receptive field centers of the nodes:

$$\begin{aligned} M_c(i,k) \quad &= w^{IED}_{i,c} \cdot \exp\left(-\frac{(IED - IED_{i,c})^2}{(\sigma^{IED}_{i,c})^2}\right) && (2)\\ &+ w^{IID}_{i,c} \cdot \exp\left(-\frac{(IID - IID_{i,c})^2}{(\sigma^{IID}_{i,c})^2}\right) && (3)\\ &+ w^{ITD}_{i,c} \cdot \exp\left(-\frac{(ITD - ITD_{i,c})^2}{(\sigma^{ITD}_{i,c})^2}\right) && (4)\end{aligned}$$

Responses are additive for each cue, therefore missing or inaccurate cues will not impair localization if the remaining cues are working properly. After node responses have been computed, the nodes with the highest responses are taken as candidate positions for the measured sound event. With $M^{max}_c(k) = \max_i(M_c(i,k))$ as the maximum response over all nodes we compute the normalized response $N_c(i,k)$:

$$N_c(i,k) = \exp\left(\frac{M_c(i,k) - M^{max}_c(k)}{\sigma_N}\right) \quad , \tag{5}$$

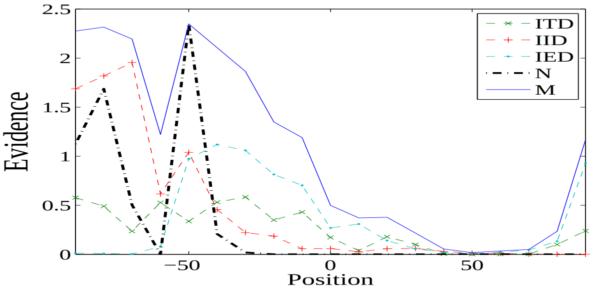

Fig. 2. Localization responses for individual cues and combined over all channels(M). The final normalized responses N are shown in black.

with $\sigma_N = 0.1$ as a normalization constant. This operation is a weak winner-take-all strategy leaving only a few strongly activated nodes. We set all responses below a threshold level $\Theta_S = 0.1$ to zero. As a result we get one or a few candidate nodes (= positions) for every recorded sound event. Cue ambiguities, e.g. as known for ITD in the high-frequency range, can be resolved via the integration of the other cues or represented as multiple location candidates if a disambiguation is not possible. Fig. 2 shows an example for a single auditory event (measured IED, IID, ITD triple). The graph plots the localization responses for the three cues individually and as a combination (M), plus the normalized responses N. Cue ambiguities are reduced to the two most likely candidate positions in this example.

D. Spectral Subtraction

An important requirement for our system is that it can suppress localization of permanently active noise sources such as fan noise and exclude the interfering source characteristics from the computation of the target cues. In order to achieve this we use the biologically plausible approach of subtracting the estimated mean value of the fan noise envelope $n^{l/r}(c,k)$ from the overall left (l) and right (r) envelope $l^{l/r}(c,k)$, i.e.

$$s^{l/r}(c,k) = l^{l/r}(c,k) - \mathrm{E}\left\{ n^{l/r}(c,k) \right\} \ . \tag{6}$$

This approach, which is known as *Spectral Subtraction* when applied in the Short Time Fourier Transform domain [14], proves to be beneficial for cue computation as it removes a strong bias. For example instead of the incorrect IID value

$$\frac{s^l(c,k) - s^r(c,k) + n^l(c,k) - n^r(c,k)}{\max(s^l(c,k) + n^l(c,k), s^r(c,k) + n^r(c,k))} \tag{7}$$

the system now calculates

$$\frac{s^l(c,k) - s^r(c,k) + \tilde{n}^l(c,k) - \tilde{n^r}(c,k)}{\max(s^l(c,k) + \tilde{n}^l(c,k), s^r(c,k) + \tilde{n}^r(c,k))} \tag{8}$$

which increases the robustness to noise as the noise mean is removed and in the case of completely deterministic signals ideal compensation for the noise is achieved. The term $\tilde{n}^{l/r}$ contains the residual zero mean noise because a statistical description is more realistic.

The remaining problem of noise level estimation is solved on-line by exploiting the fact that the mean noise value does not change quickly in time and that it can be observed solely in speech / sound event pauses. Figure 3 depicts the situation. At the beginning of the recording the microphones pick up fan noise only, then a speaker is active and speech components superimpose the noise level. However it is important to see that pauses occur naturally in speech and the sound level drops to the noise level in normal conversations. From this observation we can derive our algorithm which is a simplified filter bank adapted version of Cohen's *Minimum Controlled Recursive Averaging* [15], [16].

First, we choose a first order recursive filter structure for estimation of the mean noise envelope (note that for notational convenience we dropped the channel index c and the distinction between left and right),

$$\hat{n}(k) = \gamma(p(k))\hat{n}(k-1) + (1 - \gamma(p(k))) \cdot l(k) \ , \tag{9}$$

and make the filter's smoothing constant γ dependent on the speech probability $p(k)$ in channel c at time k:

$$\gamma\left(p(k)\right) = \gamma_{\text{min}} + (1 - \gamma_{\text{min}}) \cdot p(k) \tag{10}$$

In times of high speech probability $p(k) \approx 1$ the estimation of the mean value freezes ($\gamma = 1$) while in pauses a minimum value γ_{min} is applied which is a compromise between adaptation speed and error variance. Too high values lead to slow convergence whereas small values lead to fluctuations in the level.

In the next step we need to approximate the speech probability $p(k)$. The trick here is to use the running minimum $l_{\text{min}}(k)$ of a smoothed version of the signal envelope

$$\begin{aligned} l_s(k) &= \gamma_s \cdot l_s(k-1) + (1-\gamma_s) \cdot l(k) \\ l_{\text{min}}(k) &= \min\{l_s(m) \mid k - L + 1 < m < k\} \end{aligned}$$

as a noise baseline. Against this baseline we can then test

$$\Lambda(k) = l_s(k)/l_{min}(k)$$

and decide for speech if $\Lambda(k)$ is above a certain threshold value T_{speech}. With this hard indication of speech, i.e. $p(k) \in \{0,1\}$, we then control the averaging. However, this Voice Activity Detection scheme has the drawback that it can not respond fast to noise level changes as a long minimum filter length L is required to prevent increases during speech [15]. Therefore, a second iteration of minimum filtering is applied where detected speech segments of the first iteration are excluded, for details see [15]. The threshold values T_{speech} are obtained from simulations with fan noise only and result in different values for each channel as the bandwidth increases along the frequency axis.

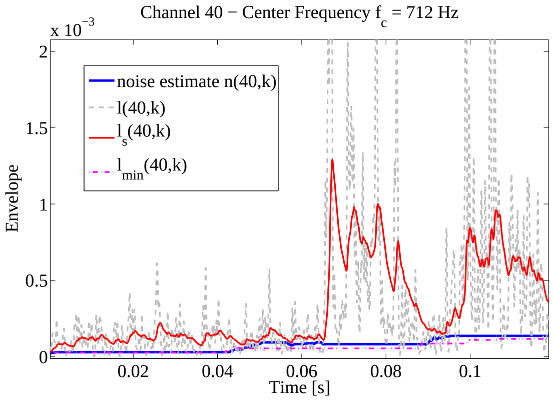

Fig. 3. Estimated mean noise level (blue) in channel 40. Note that the system is able to adapt in the short speech pause at 0.09 sec.

The complete behavior of the algorithm can be seen in figure 3. At the beginning the noise level (blue line) adjusts starting from near zero to the current level at the beginning and stops the estimation as the first spoken words arrive. Later on it resumes operation in the small pause and stops again.

Due to the noise reduction, the IID computation, which turned out to the best single cue for our scenarios, is far less affected by static noise than in competing approaches making the noise reduction an important requirement for operation in noisy conditions. Furthermore, the noise estimation also plays an important role for measuring ITD and IED values as a robust adapting noise level baseline is needed to reject cue measurement points for pure noise signals [6].

E. Calibration

The relation between cue values and positions is learned offline in a special calibration scenario where we present a number of auditory stimuli from a fixed speaker and record from our microphones while the head is moved to different defined positions. The recorded sound files are sent through the architecture to measure the cues. For the cues measured at one position we compute the mean cue values (the receptive field) plus the variance of measurements. The latter is used to assign confidence values w (see above) as is described in [6]. The receptive field width σ is set per channel depending on the range of measured cue values.

F. Integration

Evidence $E(i,k)$ for different positions (nodes) is computed by integrating normalized responses $N_c(i,k)$ over time and all frequency channels. First we integrate over time in a neuron-like fashion:

$$E_c(i,k) = \alpha * E_c(i,k-1) + N_c(i,k). \tag{11}$$

The constant α is given by $\alpha = \exp\left(-\frac{\Delta k}{\tau}\right)$, with an integration time constant $\tau = 100$ ms. Then we sum node responses over all channels:

$$E(i,k) = \sum_c E_c(i,k) \tag{12}$$

Performance can be improved considerably by smoothing the evidences over time and positions. We employed a Gaussian smoothing filter with a width of 400 ms in time and 10 degrees in positions. The result is a smoother evidence which results in better localization performance due to the integration of more localization cues for every time step and position.

G. Stream Tracking

Following the evidence computation auditory objects have to be identified and tracked over time. This process is called Auditory Streaming. For the scenario we have chosen, it suffices to track only a single auditory stream. To start a stream the maximum evidence E_{max} has to exceed a threshold T_{start}. As long as the evidence stays above T_{stop} the stream is kept active. The position of a stream x_s is initialized as:

$$x_s(k) = P(i_{\text{max}} = \arg\max_i(E(i,k))), \tag{13}$$

where $x = P(i)$ is the function that maps node indices to positions. Positions are updated as long as the stream is active by first computing the position $x_l(k)$ of the local evidence maximum. The new maximum is searched for only in the local surrounding of the current stream position ($\Delta i = 20$ degrees in system) to stabilize the search process. Then we update the stream's position by:

$$x_s(k) = \beta \cdot x_s(k-1) + (1-\beta) \cdot x_l(k) \quad . \tag{14}$$

The constant β is a smoothing parameter. In case the global evidence maximum exceeds the local maximum by a certain factor T_{switch} ($E_{\text{max}} > T_{\text{switch}} \cdot E_l$), the stream's position is instantly switched to the position of the global maximum. If E_{max} falls below T_{stop} the stream is closed.

H. Head Control

An active stream will trigger a head movement to face the perceived location of the sound source $x_s(k)$. If no streams are active, the head is kept still. Head movements and sound localization are synchronized in a way to ensure that noise generated by ego motion is suppressed. This is done by setting all $N_c(i,k)$ to zero from the time on a new head motor command has been sent up to the point where the head motion is finished. Being able to localize sound sources during head saccades is still an open problem under investigation.

III. Implementation

We implement different processing elements (e.g. filter-banks, noise reduction, temporal integration) in separate modules. The total number is more than 100 in our application. Modules are written in a standardized component model (BBCM [17], [18]). Therefore integration of modules from different researchers was comparatively easy and straightforward. The linking of modules on the software side is done within a real-time middleware and integration (called RTBOS [17], [18]) that interconnects modules flexibly and also allows the distribution of processing over several threads, CPUs and even computers. Network communication is done via TCP/IP. As a result we can flexibly integrate a large number of modules (see also [19] for another large-scale system using the BBCM/RTBOS system) for sequential and parallel execution. To speed-up computation we also make use of Intel's IPP library.

A. Recording Hardware

Two DPA 4060-BM omni-directional microphones and a MAudio Delta1010 recording system are used to record sound data with a sampling rate of 24 kHz. The microphones are mounted on different humanoid heads at the approximate positions of human ears. Heads are filled with foam or other damping material but are otherwise basically empty. The head is mounted on a neck element (Amtec Robotics PowerCube PW070), connected to a PC via CANbus. The head can turn 360 degrees at high speed, which is unfortunately accompanied by substantial noise (due to the close proximity of the neck to the microphones).

B. Timestamp-based Synchronization

To optimize processing speed vs. communication overhead sound data is analyzed in blocks of 50 ms length (1200 samples). As different parts of the system can run in parallel, a synchronization of data blocks is necessary. We use a timestamp generated in the sound recording module that is transferred by RTBOS throughout the system to align blocks from different processing streams and to detect holes in the processing chain. Our system can handle missing blocks and will even show an acceptable performance in case of frequent discontinuities (see Fig. 4). These are normally the result of either network communication delays or high computational load for some CPUs.

C. Latency Compensation and Subsampling

We also compensate processing latencies that result from using different filters for cues and measurement windows: ITD computation works on the direct output of the Gammatone filter bank while the measurement window is calculated on a low-pass-filtered envelope signal. Filters introduce a group delay which leads to different latencies. These can sum up to 624 samples (26 ms). If this difference is not compensated, cues will be measured outside the optimal window (see [6]) leading to severe impairments in echoic environments. Our architecture can handle arbitrary latency differences and compensates them when needed by delaying the faster signal. The compensation is done per channel so that the low-latency high-frequency channels are not blocked by the high-latency low frequency channels. This operation is executed together with the timestamp synchronization, as depicted in Fig. 4.

As can be seen in the system graph (Fig. 1), signals are downsampled at different stages of processing. The subsampling factor used is 24, which means that a large share of the system effectively runs at 1 kHz only. This results in a speed up of approximately 2 for the overall system.

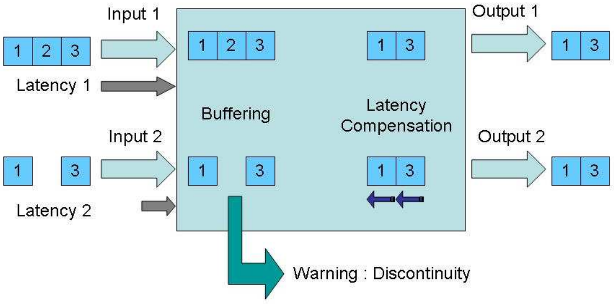

Fig. 4. The operation of the synchronization module for the example of two inputs. Blocks are stored internally until all inputs with a common timestamp have been received. A missing block in at least one input will trigger a warning message and the data in this block will be skipped. After synchronization, latencies are compensated by delaying the faster signal (here input 2).

IV. Results

Our architecture was tested on-line in two different rooms. Both rooms were noisy and echoic (750 ms and 330 ms reverberation time), but the system showed a robust localization in all cases. In an on-line scenario several people attracted the system's attention by calling it or making different sounds. Even in this very noisy environment the system found the correct sound source using at most three, but normally only a single head movement. Localization performance was still good when music was played and very strong background noise (Asimo fan noise) was added. More information on these scenarios and some results can be found in [6]. The system also shows a quick response, we measured a response latency of less than 400 ms (between signal onset and generation of a head movement command). Now we present some results of our system working offline on pre-recorded soundfiles.

A. Low-noise Scenario

The system runs on-line either stand-alone on a single machine using 60 channels and with an additional 4-CPU-SMP with up to 180 channels of the GFB. We investigated the effect of increasing the number of frequency channels by testing the performance of the system on a database recorded with one of our heads. Sound files were recorded at 1 degree increments. We used 20 files for training and 15 files (short human (English) utterances) for testing. Data was recorded in a normal echoic room (7x15 m, reverberation time 750 ms) with a modest level of background noise (air condition, computer fan). The SNR of test files was around 10-15 dB. Microphones were mounted on a humanoid dummy head. We evaluated the mean localization error $\bar{\epsilon}$ (in degrees) over all test files and positions, the maximal deviation from the true target over all files ϵ_{max}, and the mean localization error per channel $\bar{\epsilon_c}$ (in degrees). The range of source positions is 180 degrees (-90 to + 90 degrees). The results are given in the following table:

channels	$\bar{\epsilon}$	ϵ_{max}	$\bar{\epsilon_c}$
30	2.8°	46°	17.8°
60	2.3°	42°	17.2°
90	2.2°	43°	17.0°
120	2.2°	42°	16.9°
180	2.2°	39°	16.8°

This experiment shows that there is an improvement in performance with increasing number of channels, but only marginally when going beyond 60 channels. As can be seen the system has a very high precision for high SNR speech signals in normal environments. Outliers are very rare (more than 99% of all sound sources are localized within 10 degrees of the correct position) and in almost 20% of all cases the localization had a precision of 1 degree. The mean error per channel, though, is considerably higher, clearly demonstrating the need to integrate over channels.

B. Test-case results under different noise and echo conditions

We also performed tests on data recorded from Asimo's head microphones (data kindly provided by K. Nakadai) in two different rooms (one anechoic and a normal office room with considerable echoes). We also compared the situation where Asimo is turned off (SNR of ca. 12 dB) and turned on (approximately 6 dB SNR). The system was tested with a fixed setting (60 channels of the Gammatone filterbank, 24 kHz sampling rate) to investigate the effect of noise and echo. Data was recorded at positions spaced 10 degrees apart from -90 to +90 degrees. We got the mean integrated localization error / mean channel-wise localization error for the different scenarios as follows:

$\bar{\epsilon}$ / $\bar{\epsilon_c}$	Asimo off	Asimo on
anechoic	0.39°/ 15.36°	2.46°/ 39.77°
echoic	1.96°/ 28.26°	1.51°/ 35.71°

As can be seen the performance in a noise- and echo-free environment is almost perfect. Echoes reduce performance not as much as Asimo's strong fan noise. The combination of echo and noise is less severe than noise alone in this case which is due to a few outliers in the data. Fig. 5 shows the mean localization error per channel for the four different scenarios. The highest frequency channels are generally poorly performing and especially channels between 600 Hz and 3 kHz (channel number 18 to 40) are strongly affected by noise. Despite this increase in mean channel error, the overall performance is still very good, considering the high noise level.

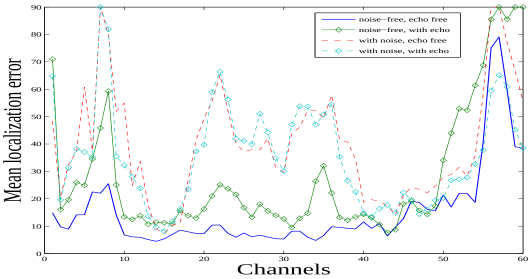

Fig. 5. Mean localization error per channel for different noise and echo conditions.

V. Summary and Conclusion

We have presented a system for real-time, real-world sound source localization using standard hardware and a two-microphone set-up on a humanoid head. The system shows a robust performance even under noisy and echoic conditions. This robustness is achieved by taking inspiration from biological auditory processing systems. Firstly, we are using a biologically-inspired integration of cues. Secondly we model the precedence effect for echo-cancellation [6] which improved the stability and reliability of cue computation considerably. Finally, we are using the Gammatone-Filterbank instead of FFT and zero-crossings instead of correlation-based approaches for IED and ITD computation. It is therefore demonstrated that biologically inspired real-time sound localization in an every-day environment can be achieved with conventional hardware and using just two microphones on a humanoid head.

The current architecture will be the basis for the integration of additional auditory processing capabilities, like e.g. pitch tracking [20], which will require a larger number of frequency channels and more processing modules to operate. The presented architecture has the capacity to be expanded to meet these requirements

Acknowledgment

We would like to thank Marcus Stein, Mark Dunn and Antonello Ceravola for their support in building this system. We also thank Volker Willert and Julian Eggert for fruitful discussions. Special thanks to Kazuhiro Nakadai for providing us with the Asimo test data and help in the initial stage of this work.

References

[1] H. G. Okuno, K. Nakadai, T. Lourens, and H. Kitano, "Sound and visual tracking for humanoid robot." *Appl. Intell.*, vol. 20, no. 3, pp. 253–266, 2004.

[2] K. Nakadai, H. Nakajima, K. Yamada, Y. Hasegawa, T. Nakamura, and H. Tsujino, "Sound source tracking with directivity pattern estimation using a 64 ch microphone array," in *Proc. Int. Conf. Intelligent Robots and Systems (IROS) '05*, Edmonton, Canada, 2005, pp. 196–202.

[3] S. Kurotaki, N. Suzuki, K. Nakadai, H. G. Okuno, and H. Amano, "Implementation of active direction-pass filter on dynamically reconfigurable processor," in *Proc. Int. Conf. Intelligent Robots and Systems (IROS) '05*, Edmonton, Canada, 2005, pp. 515–520.

[4] E. Berglund and J. Sitte, "Sound source localisation through active audition," in *Proc. Int. Conf. Intelligent Robots and Systems (IROS) '05*, Edmonton, Canada, 2005, pp. 509–514.

[5] S. Kagami, Y. Tamai, H. Mizoguchi, K. Nishiwaki, and H. Inoue, "Detecting and segmenting sound sources by using microphone array," in *Proceedings of 2004 IEEE-RAS/RSJ International Conference on Humanoid Robots(Humanoids2004)*, 11 2004, pp. 67 paper(CD–ROM).

[6] M. Heckmann, T. Rodemann, B. Schölling, F. Joublin, and C. Goerick, "Binaural auditory inspired robust sound source localization in echoic and noisy environments," in *Proceedings of the International Conference on Intelligent Robots & Systems (IROS)*. IEEE, 2006, p. submitted.

[7] N. Roman, D. Wang, and G. J. Brown, "Speech segregation based on sound localization," *J. Acoust. Soc. Am.*, vol. 114, no. 4, pp. 2236–2252, 2003.

[8] S. Yamamoto, K. Nakadai, J.-M. Valin, J. Rouat, F. Michaud, K. Komatani, T. Ogata, and H. G. Okuno, "Making a robot recognize three simultaneous sentences in real-time," in *Proc. Int. Conf. Intelligent Robots and Systems (IROS) '05*, Edmonton, Canada, 2005, pp. 897–902.

[9] M. Slaney, "An efficient implementation of the patterson-holdsworth auditory filterbank,," Apple Computer Co., Technical Report 35, 1993.

[10] Y.-I. Kim, S. J. An, R. M. Kil, and H.-M. Park, "Sound segregation based on binaural zero-crossings," in *Proc. Int. Conf. on Spoken Lang. Proc. (ICSLP) 05*, Lisboa, Portugal, 2005, pp. 2325–2328.

[11] V. Willert, J. Eggert, J. Adamy, R. Stahl, and E. Körner, "A probabilistic model for binaural sound localization," *IEEE Transactions on Systems, Man and Cybernetics - Part B, accepted*, 2006.

[12] R. D. Patterson, K. Robinson, J. Holdsworth, D. McKeown, C. Zhang, and M. H. Allerhand, *Auditory Physiology and Perception*. Exford: Pergamon, 1992, ch. Complex sounds and auditory images, pp. 429–446.

[13] B. R. Glasberg and B. C. J. Moore, "Derivation of auditory filter shapes from notched-noise data," *Hearing research*, vol. 47, pp. 103–108, 1990.

[14] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 27, no. 2, pp. 113–120, 1979.

[15] I. Cohen, "Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 5, pp. 446–475, September 2003.

[16] I. Cohen and B. Berdugo, "Noise estimation by minima controlled recursive averaging for robust speech enhancement," *IEEE Signal Processing Letters*, vol. 9, no. 1, pp. 12–15, January 2002.

[17] A. Ceravola and C. Goerick, "Towards designing real-time brain-like computing systems," in *The First International Symposium on Nature-Inspired Systems for Parallel, Asynchronous and Decentralised Environments (NISPADE 2006), Bristol, England, accepted*, 2006.

[18] A. Ceravola, F. Joublin, M. Dunn, J. Eggert, M. Stein, and C. Goerick, "Integrated research and development environment for real-time distributed embodied intelligent systems," in *Proceedings of the International Conference on Intelligent Robots & Systems (IROS)*. IEEE, 2006, p. submitted.

[19] C. Goerick, H. Wersing, I. Mikhailova, and M. Dunn, "Peripersonal space and object recognition for humanoids," in *Proceedings of the IEEE/RSJ International Conference on Humanoid Robots (Humanois 2005), Tsukuba, Japan*, 2005.

[20] M. Heckmann, F. Joublin, and E. Körner, "Sound source separation for a robot based on pitch," in *Proceedings of the International Conference on Intelligent Robots & Systems (IROS)*. IEEE, 2005, pp. 203–208.