Auditory Robotic Tracking of Sound Sources using Hybrid Cross-Correlation and Recurrent Networks

John Murray, Stefan Wermter, Harry Erwin
Hybrid Intelligent Systems
University of Sunderland
Sunderland, England, SR6 0DD
{john.murray,stefan.wermter,harry.erwin}@sunderland.ac.uk

***Abstract* –This paper describes an auditory robotic system capable of computing the angle of incidence of a sound source on the horizontal plane (azimuth). The system, with the use of an Elman type recurrent neural network (RNN), is able to dynamically track this sound source as it changes azimuthally within the environment. The RNN is used to enable fast tracking responses to the overall system over a set time, as opposed to waiting for the next sound position before moving. The system is first tested in a simulated environment and then these results are compared with testing on the robotic system. The results show that the development of a hybrid system incorporating cross-correlation and recurrent neural networks is an effective mechanism for the control of a robot that tracks sound sources azimuthally.**

Index Terms – Sound source, tracking, robotics, prediction, cross-correlation.

I. Introduction

Due to the increasing developments in robotics, robots are becoming more common in everyday situations. In order for robots to be able to operate safely within the environment it is necessary for them to be able to perceive their surroundings. Over the years research has been conducted in different areas using various modalities to achieve the goal of robotic environment perception. Such modalities have included vision [1, 2], sonar [3], laser [4] and more recently acoustics [5].

Acoustics can provide advantages over vision, sonar and laser modalities to aid in the tracking of sound sources. The main advantage of incorporating acoustics into the system to perceive the environment is that sound can travel 'around' objects due to reflection and reverberation off walls and surrounding objects, whereas other modalities are unable to achieve this. For example, a visual system would be unable to detect an object around the corner of a corridor due to it being out of its line of sight. The disadvantage however of using a modality such as audition is that any object to be tracked *must* have an acoustic element.

In order for robots to integrate more easily into everyday life, and for them to not only be widely accepted but also be easier to communicate with, it is therefore necessary to develop robotic systems that interact in the same manner as humans. Extensive research has been conducted into the sociable aspects of robots, such as expressions [6], speech (understanding and perception) [7] and movement [8].

In this paper we focus on auditory robotic tracking of sound sources, as this is an essential task for sound and speech processing in order to improve the signal to noise ratio of the sound. Inspiration for our research has been taken from the mammalian auditory system. Mammals have excellent auditory capabilities, with humans reaching an accuracy of $\pm 1^{o}$ azimuth and $\pm 5^{o}$ elevation [9]. We see mechanisms within the auditory cortex of the human that provide the ability for the localisation of sounds; these include the Jeffress model [10, 11] which shows the use of coincidence detectors that are trained to determine Interaural Time Differences (ITD) between the two ears. Our system has therefore been developed with the human auditory system in mind, and hence has been built with only two 'ears' and draws on auditory cues that exist in biology for azimuth estimation [10].

This paper is structured as follows: in section II we explain the initial stage of our model, including how it is used in the overall system, section III explains the neural part of our architecture including training and testing, and section IV describes how the individual components of the model join to form the hybrid architecture.

The system model has been developed in two main stages, the first of which is known as the Azimuth Estimator [12] and is used to determine the initial angle of incidence of the sound source relative to the robot. The second main stage is the Neural Speed Estimator and Predictor. This stage includes the RNN that has been used for the overall system model and is used to compute the relative speed of the sound source within the environment. The RNN informs the robot to attend to the next *predicted* azimuth position of where the sound source is expected to be within the environment. The prediction is based on the speed of the sound source, and is determined by the increment in angles per time step of the movement of the sound source.

Together these two stages are combined to provide a dynamic robot tracking system that can vary its speed and position with respect to the stimulus received from the object in question. This model has been developed with the idea of a robotic service waiter in mind; this robotic waiter would listen for commands and then attend to the person issuing those commands. As the person moves within the environment the robot would need to maintain an acoustic track for the purpose of improving the signal-to-noise (SNR, which is a reduction in

the background noise relative to the sound emanating from the sound source), so as to enable more accurate estimation of direction.

II. Azimuth Estimation

This initial stage of the system is used to detect the azimuth position of the sound source and in turn feed this value into the neural processing Speed Estimator and Predictor stage (RNN). The azimuth represents the horizontal angle of incidence of the sound source with respect to the robot's internal frame of reference, which is head-centred. To determine the azimuth we need to calculate the time delay of arrival (TDOA). The TDOA is the time difference between the sound arriving at the ipsilateral and contralateral microphones. This is the equivalent in biological terms of the mammalian ITD [11] discussed earlier. We chose to utilise this particular auditory cue because the same constraints and signals that are present in the human auditory system are also available to our robotic system.

We compute the TDOA using a method known as cross-correlation, which has also been shown to exist biologically [13]. The signal processing version of the cross-correlation method presented in our model (see equation 1) is used to compare two signals *g(t)* and *h(t)* for the delay offset **σ** (i.e. the amount of time samples between the points where the two signals are at maximum similarity). Fig. 1 shows the two signals *g(t)* and *h(t)* being analysed for their maximum similarity via the cross-correlation method, the shaded area within Fig. 1 shows the positions within the signals where maximum similarity occurs.

Here our two signals *g(t)* and *h(t)* are the sounds received at the left and right microphones during the current recording. From the cross-correlation of *g(t)* and *h(t)* a product vector ***C*** is created where the maximum value within this vector is the point of maximum similarity between the two signals. Fig. 2 shows this correlation vector.

The system uses finite time sequences of data to compute the cross-correlation of the two signals. For each sound recording a period of 200ms of sound is recorded for presentation to the cross-correlation function. With a sample rate of 44.1 KHz this gives an N × M matrix of 2 × 8820 where each value within the matrix represents the amplitude value (between ±1) of the sound perceived in the environment.

$$Corr(g,h)_j(t) \equiv \sum_{k=0}^{N-1} g_{j+k} k_k \qquad (1)$$

As the sounds are recorded with a sample rate of 44.1 KHz, each delay (i.e. each sample) within the correlation vector ***C*** has a time increment Δ of 22.67μs. To enable us to calculate the azimuth of the sound source we need to determine the delay **σ** from the correlation vector ***C***; that is, by how many samples the two signals are offset.

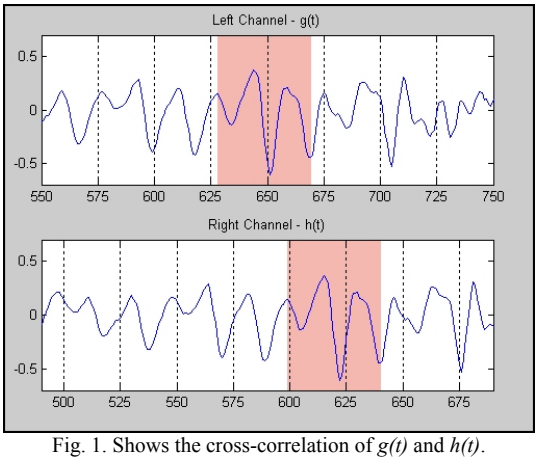

Fig. 1. Shows the cross-correlation of *g(t)* and *h(t)*.

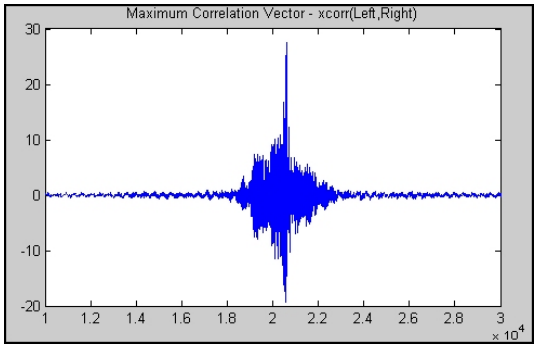

Fig. 2. Shows the resultant cross-correlation vector ***C***.

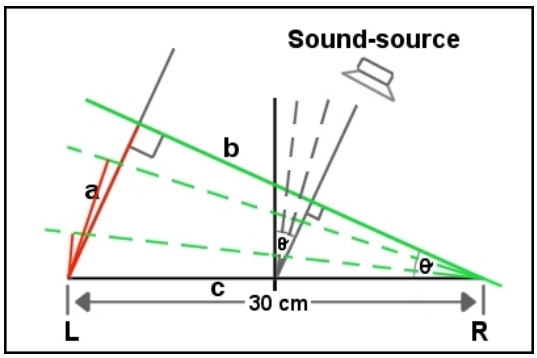

Fig. 3. Trigometric triangle for calculating the azimuth.

If we assume the sound source is at 0° then the signals received at either microphone will be in phase due to them arriving at the same point in time. Cross-correlation would therefore provide a delay **σ** of 0 due to the positions of maximum similarity within the signals *g(t)* and *h(t)* being in the same positions within the wave forms. In order to produce the vector ***C*,** cross-correlation offsets the two signals to their furthest points and then 'slides' these across each other in a sliding window effect until they are offset in the opposite direction, resulting in a vector whose size is defined by equation 2. The maximum point within ***C*** for an angle of 0° occurs at the midpoint of ***C*,** therefore the delay offset for two signals is calculated by equation 3.

$$\text{length}(\boldsymbol{C}) = (\text{length}(g(t)) + \text{length}(h(t)))-1 \qquad (2)$$

$$\boldsymbol{\sigma} = \text{Length}(\boldsymbol{C})/2 - \boldsymbol{C}_{MAX} \quad (3)$$

where $\boldsymbol{C}_{MAX}$ is the position in $\boldsymbol{C}$ which contains the largest value from the product of the elements being correlated.

Using basic trigometric functions we can now determine the angle $\boldsymbol{\Theta}$. Firstly we need to decide which sides 'a', 'b' or 'c' we have to utilise to find the angle $\boldsymbol{\Theta}$. From trigonometry we have the following equations:

$$Sin\Theta = \frac{a}{c}, Cos\Theta = \frac{b}{c}, Tan\Theta = \frac{c}{b} \quad (4)$$

In order to determine the angle $\boldsymbol{\Theta}$ within Fig. 3 we need to know the length of at least two of the sides. We already know the length of 'c' which is set at a constant of 0.30 metres, which is the distance between the two microphones. From the remaining two sides we can now use the TDOA to help compute the length of side 'a', as the TDOA is proportional to the length of 'a' (i.e. the larger the angle of incidence the greater the TDOA and the larger 'a' becomes), see equation 5.

$$t = \Delta \times \sigma \quad (5)$$

where Δ = time between sound samples (i.e. 22.67μs) and σ = the number of delay samples returned from the cross-correlation function, equation 3.

Next, we determine the length of 'a' by substituting equation 5 into the following equation:

$$length(a) = t \times V_{sound} = (\Delta \times \sigma) \times V_{sound} \quad (6)$$

where speed of sound is taken to be v = 345m/s at room temperature of 22°C @ sea level.

We now know the length of sides 'a' and 'c', therefore (referring to equation 4) the trigometric function required in order to determine $\boldsymbol{\Theta}$ is that of the sine rule.

$$Sin\Theta = \frac{a}{c} \quad (7)$$

Transposing equation 7 therefore gives:

$$\Theta = Sin^{-1}\frac{a}{c} \quad (8)$$

The final result from equation 8 gives us the azimuth position of the sound source within the environment. This is next passed to an algorithm for further processing, see Fig. 6. The initial stage of this model provides us with the ability to calculate the azimuth of a sound source within the external environment by using cross-correlation to estimate the TDOA of two signals.

III. Neural Speed Estimator and Predictor

The goal of the Neural Speed Estimator and Predictor stage is to provide a predicted 'next position' of the moving sound source for the robot to attend to. The main purpose of the predicted position is to enable the system to move to the next position of the sound source when no sound signal is detected; that is, if the sound source is occluded by an obstacle or stops transmitting for a period then the system can still temporally track the position of the source, therefore providing a more real-time based implementation of a robotic tracker. The Neural Speed Estimator and Predictor stage is designed using a recurrent neural network (RNN) and receives its input in the form of activation of a specific neuron on its input layer, representing the current angle of the sound source from the initial starting position of 0°. The activation of the correct neuron is determined by equation 9 which is based on several variables contained within the system. These variables contain current position (from starting point of 0°), distance moved and perceived angle of sound source (angle from robot's head centred coordinate frame). The activations are used by the RNN to determine the next predicted value within the temporal sequence for the motion of the sound source. The network is a recurrent architecture, and therefore provides the ability to recognise patterns through time and hence learn temporal sequences.

$$\text{IP activation} = \text{Current position} + \text{Angle Perceived} \quad (9)$$

Within the network architecture there are four separate layers that make up the RNN, these are:

Layer 1 – Input – 45 Units
Layer 2 – Hidden – 30 Units
Layer 3 – Context – 30 Units
Layer 4 – Output – 45 Units

The network architecture is based on that of an Elman network [14] where the hidden unit activations are used to provide previous context information to the network. This context information is acquired by providing one-to-one connections between the hidden and context layer, these connections are provided with weight values of 1.0 to prevent biasing the context activations. The one-to-one projections are used to copy the hidden layer activations to the context layer; this is required to ensure that the context of the previous pattern at time step t_{-1} is still available to the network at time step t, see equation 11.

Our network adopts the standard backpropagation algorithm for training and weight updating, see equation 10. The stopping criterion is based on the sum of the squared error (SSE), i.e. the difference between the output activations versus the desired target activations (provided by the training environments). Once the SSE reaches a value of 0.04 (meaning that between two successive epochs the SSE tolerance

must be within this value) then the network is said to have converged to within the accepted tolerance level.

$$\Delta w_{ij}(n+1) = \eta \delta_j o_i + \alpha \Delta w_{ij}(n) \qquad (10)$$

where Δw_{ij} = the weight change between neurons i and j, n = current pattern, η = momentum = 0.25, δ_j = error of neuron j, o_i = output of neuron i, α = momentum term to prevent the weight update entering oscillation by adding a small amount of the previous weight change to the current weight change.

$$a_i^C(t+1) = a_i^H(t) \qquad (11)$$

To ensure the system correctly classifies temporal patterns each pattern is stored in a separate sub-group training environment. The events (input sequences, i.e. angle representations) within each sub-group are then presented sequentially to the network, whilst each sub-group is presented randomly. This prevents the network from learning the temporal sequences of the actual sub-groups themselves whilst still learning the sequences within the sub-groups.

Fig. 5 shows the first sub-group training environment for speed 1. As can be seen the events are presented in a sequential manner. On the presentation of Event_0 there is activation on the first input neuron, however there is no output activation.

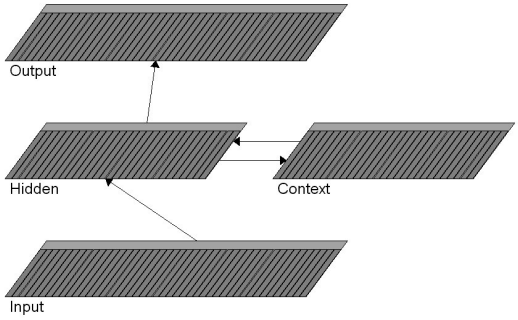

Fig. 4. Recurrent Neural Network architecture used in model.

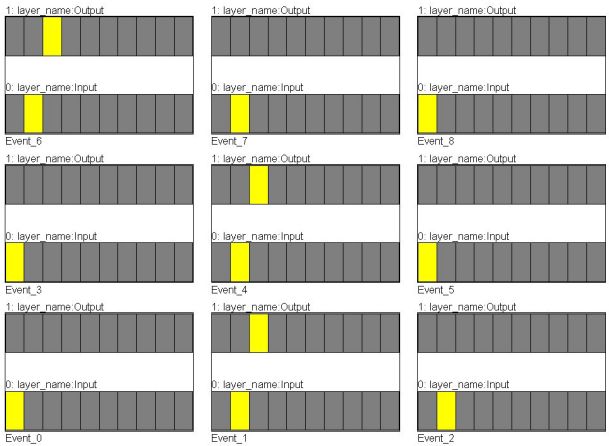

Fig. 5. Sample of Speed 1 training events.

Table 1. Representation of the input activations for the first five speeds.

Speed	Input Representation
1_{t1}	1000000000……..
1_{t2}	0100000000……..
2_{t1}	1000000000……..
2_{t2}	0010000000……..
3_{t1}	1000000000……..
3_{t2}	0001000000……..
4_{t1}	1000000000……..
4_{t2}	0000100000……..
5_{t1}	1000000000……..
5_{t2}	0000010000……..

This is due to Event_0 representing the first step in the temporal sequence (shown in table 1) and so the network needs to wait for the second input before the required output can be calculated as shown by Event_1. Event_7 and Event_8 differ from the first 7 events in that they are presented to the network in reverse temporal order with no output activation. This is to train the network *not* to provide activation based on just the input of Event_7 or Event_8, or on the reverse presentation of the sequence. Fig. 5 therefore shows the desired output activations for a specific temporal pattern (Note: for the purposes of this paper presentation, Fig. 5 only shows the first 10 input and output neurons from the network). 20 different speeds in total were provided to the network for training, with each sub-group being configured in the same manner (i.e. in temporal order with the last two events reversed for the above reason), giving a total of 180 training events.

The system has the ability to determine the azimuth within a 180^0 field, which is $\pm 90^o$ to the left and right of 0^o. Each input unit within the network represents 2^0 of azimuth either ± depending on the direction of motion. Upon the presentation of the initial angle to the system, the network waits for a second angle before it predicts the third. When output activation is present on the network then this is used to direct the robot to its next required position.

The network converged (i.e. completed training) after 35000 epochs, taking 20 minutes to train. The training of the network was conducted on an Intel Pentium 4 2GHz processor. Once training was complete the weights within the network between the various neurons were recorded and saved to file for use on the robot, therefore preventing training being necessary every time the system is restarted.

This second stage of our model enables the robot to provide two consecutive temporal inputs in order to obtain a third attendable goal position. This, combined with the azimuth estimator from stage 1, constitutes the main aspect of the overall system model.

IV. Hybrid Architecture

The final stage to the model is the encapsulation, control and connection of the Azimuth Estimator and Neural Speed Estimator and Predictor. The system maintains several variables as previously discussed to accomplish the goal of sound source tracking. As the robot maintains its own coordinate

reference, detected angles are referred to as perceived due to world coordinates being set at the initial starting position. These variables enable the system to maintain its track on the dynamically moving source whilst providing inputs to the RNN to enable prediction.

The neural network requires two sequential input activations at time t_0 and t_1 before the predicted estimation of the next position at t_2 can be made, therefore the system has been designed only to attend to the output of the neural network when output activation has a value greater than 0.9 on any one of the output neurons. Figs. 7 and 8 show the input and output activations of two time sequential events presented to the network.

As can be seen from Fig. 7, input neuron 1 has activation at time t_0. However, there is no output activation response due to this being the first input pattern, and hence there is no valid sequential temporal pattern. At time t_1 the second input activation is presented to the network, and as can be seen from Fig. 8 output activation is now given by the network as a valid sequential sequence has been provided. The robot will therefore be instructed to attend to this output activation position, represented by the specific neuron. However, if no output activation is received from the network the system uses the algorithm defined in Fig. 6 to instruct the robot to attend to the position the sound was detected at from the cross-correlation function.

```
At initialisation: set all variables to 0
    Loop (until exit of system)
        if output activation on RNN > 0.9
            move to RNN output activation position
            - currentPos
        else
            move to PAngle
        sample environment for sound
            update variables
            RNN ip activation = PAngle + CurrentPos
    End Loop
Exit System
```

Fig 6. Shows the algorithm of the model

This brings together the separate stages of the model, providing a hybrid architecture for the detection, azimuth prediction and tracking of sound sources within the local environment.

V. Experimentation & Testing

The first stage of testing was to ensure that the network had trained correctly to provide the desired responses from the various input activations available to the system. Once the network had converged from training, two testing environments were created. The testing environments were the same as the training environments, in the sense that they contained multiple events to present to the network in a sequential manner; however these environments were not used to update the weights or aid in the training of the network.

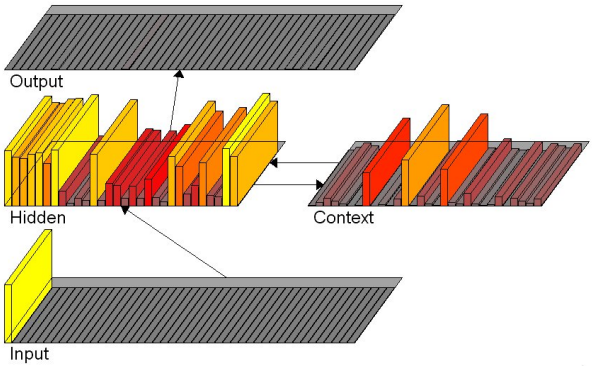

Fig. 7. Activation of IP & OP neurons at time t_0.

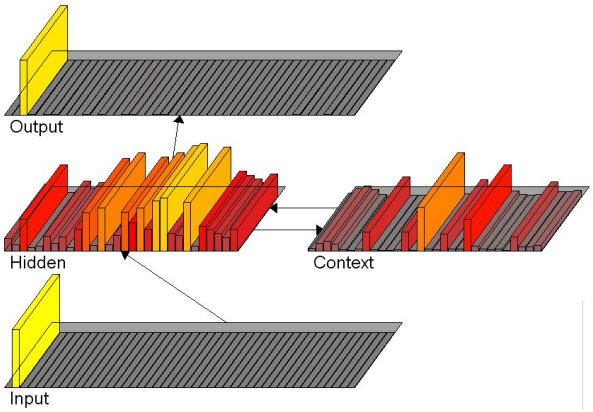

Fig. 8. Activation of IP & OP neurons at time t_1.

The testing environments also differed in that they did not provide output activations; instead they presented to the network input activations only, the outputs were then determined by the network. Test environment one contained 10,000 separate randomly generated events. The purpose of this environment was to ensure the correct response from the network when presented with unforeseen input sequences. That is, if the network receives two sequentially acceptable events then correct output activation should be provided. However, if the input patterns are not valid (e.g. two events with the same input activation) then the network should not provide any output for the system. The second training environment was manually created and consisted of 100 valid sequential events. This environment was used to test the network to ensure correct output activation based on valid sequential inputs.

The complete model was tested using pre-recorded sound files recorded from specifically measured angles within the environment. The robot was positioned within the centre of the robot lab and a speaker was positioned 1.5 metres from the centre of the robot. Forty sound files were recorded with a duration of 200ms in increments of 4.5^{o} and stored to file for presentation to the model, each of these 40 positions was recorded five times to test for accuracy and repeatability. During the worst case the model correctly categorised 33 of the sound files azimuth positions via cross-correlation and then activated the respective input neurons. After sequential input of the patterns (two at a time) the network produced output

activations. However, due to various incorrect classifications on various trials from the azimuth stage some of the sequential patterns returned incorrect prediction estimates. Table 2 shows the azimuth test trial results including the number of correctly predicted sequential events.

VI. Discussion

A model for sound source estimation has been successfully developed based on a hybrid structure. Cross-correlation has proved an excellent method for determining the TDOA [13]. The success of the cross-correlation function can be attributed to the use and application of the same auditory cues that are used in biology, such as ITD [10, 11]. The ITD of the sound cannot however be used to determine the distance away from the robot but only the angle of incidence. Other auditory cues exist which enable the estimation of distance, such as Interaural Level Difference [15], however none of these have been modelled in our system.

The RNN implemented in our model has provided the ability to recognise sound source speed based on predetermined sequential training data. Other Artificial Neural Networks (ANNs) have shown that short-term memory which retains previous knowledge is required for the prediction of time step n, therefore providing context for prediction. This has been shown in the simple recurrent networks (SRN) devised by Elman [14]. Stages could also be integrated into our model to enable online training of the system, i.e. the model could use the input activations (from three time steps t_0, t_1 and t_2) from the real world to train the network.

VII. Conclusion

In this paper we have presented a hybrid system of cross-correlation and Recurrent Neural Networks for robotic sound source azimuth estimation and prediction. The system is centred on the inspiration gained from auditory cues in the human auditory system and consists of two microphones mounted on the base of the robot, (see Fig. 9) to ensure tracking remains as real-time as possible. The hybrid architecture consists of a RNN with temporal pattern recognition that informs the robot to attend to a predicted azimuth position before the sound source itself arrives at that location.

Future work is to be conducted into enabling the system to maintain a longer predetermined silent track (e.g. when the sound source has been occluded by an object or has temporally become silenced). This is to be accomplished by introducing a decay function into the model. This decay will initialise at zero and increase by a set increment as it hears a sound until it reaches it maximum value of one. If the sound ceases, the decay will begin to gradually decrease until it again reaches zero or hears the sound again. Provided the decay value is greater than zero the system will continue to predict ahead the next location of where (based on its previous trajectory) it expects the sound source to be. It is hoped that this will provide a more realistic biologically inspired tracking model.

Table 2. Shows the test trials for azimuth and prediction classification.

Trial Number	Num of Angles	Correctly Identified	Accuracy %	Predictor Trials	Correctly Predicted
1	40	35	87.5	20	17
2	40	38	95	20	19
3	40	35	87.5	20	16
4	40	36	90	20	18
5	40	33	82.5	20	16

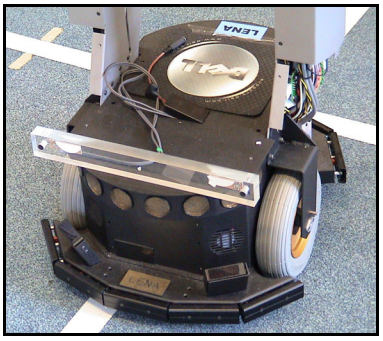

Fig. 9. Robot used for sound source tracking.

References

[1] Böhme, H.-J., et al., *An approach to multi-modal human-machine interaction for intelligent service robots.* Robotics and Autonomous Systems, 2003. **44**(1): p. 83-96.

[2] Lima, P., et al., *Omni-directional catadioptric vision for soccer robots.* Robotics and Autonomous Systems, 2001. **36**(2-3): p. 87-102.

[3] Carelli, R. and E.O. Freire, *Corridor navigation and wall-following stable control for sonar-based mobile robots.* Robotics and Autonomous Systems, 2003. **45**(3-4): p. 235-247.

[4] Matthies, L., et al., *A portable, autonomous, urban reconnaissance robot.* Robotics and Autonomous Systems, 2002. **Volume 40**(2-3): p. 163-172.

[5] Wang, Q.H., T. Ivanov, and P. Aarabi, *Acoustic robot navigation using distributed microphone arrays.* Information Fusion, 2004. **Volume 5**(Issue 2): p. 131-140.

[6] Breazeal, C., *Emotion and sociable humanoid robots.* International Journal of Human-Computer Studies, 2003. **59**(1-2): p. 119-155.

[7] Roy, D. and N. Mukherjee, *Towards Situated Speech Understanding: Visual Context Priming of Language Models.* Computer Speech and Language, Article in Press.

[8] Adams, B., *Learning Humanoid Arm Gestures.* Working Notes - AAAI Spring Symposium Series: Learning Grounded Representations, 2001: p. 1-3.

[9] Blauert, J., *Table 2.1*, in *Spatial Hearing - The Psychophysics of Human Sound Localization*. 1997. p. 39.

[10] Hawkins, H.L., et al., *Models of Binaural Psychophysics*, in *Auditory Computation*. 1995, Springer. p. 366-368.

[11] Jeffress, L.A., *A place theory of sound localization.* Journal of Computational Physiology and Psychology, 1948. **41**: p. 35-39.

[12] Murray, J.C., H.R. Erwin, and S. Wermter. *Robotics Sound-Source localization and Tracking Using Interaural Time Difference and Cross-Correlation*. in *AI Workshop on NeuroBotics*. 2004. Germany.

[13] Hawkins, H.L., et al., *Cross-Correlation Models*, in *Auditory Computation*. 1995, Springer. p. 371-377.

[14] Elman, J.L., *Finding structure in time.* Cognitive Science, 1990. **14**(Issue 2): p. 179-211.

[15] Sabin, A.T., E.A. Macpherson, and J.C. Middlebrooks, *Human sound localization at near-threshold levels.* Hearing Research, 2005. **199**(1-2): p. 124-134.