A Hybrid Architecture using Cross-Correlation and Recurrent Neural Networks for Acoustic Tracking in Robots

John C. Murray, Harry Erwin and Stefan Wermter

Hybrid Intelligent Systems
School for Computing and Technology
University of Sunderland, Tyne-and-Wear,
SR6 0DD, UK
John.Murray@sunderland.ac.uk

Abstract. Audition is one of our most important modalities and is widely used to communicate and sense the environment around us. We present an auditory robotic system capable of computing the angle of incidence (azimuth) of a sound source on the horizontal plane. The system is based on some principles drawn from the mammalian auditory system and using a recurrent neural network (RNN) is able to dynamically track a sound source as it changes azimuthally within the environment. The RNN is used to enable fast tracking responses to the overall system. The development of a hybrid system incorporating cross-correlation and recurrent neural networks is shown to be an effective mechanism for the control of a robot tracking sound sources azimuthally.

1. Introduction

The way in which the human auditory system localizes external sounds has been of interest to neuroscientists for many years. Jeffress [1] defined several models of how auditory localization occurs within the Auditory Cortex (AC) of the mammalian brain. He developed a model for showing how one of the acoustic cues, namely that of the Interaural Time Difference (ITD) is calculated. This model describes the use of neurons within the auditory cortex as coincidence detectors [2]. Jeffress also describes the use of coincidence detectors for other auditory cues, namely Interaural Level Difference within the auditory cortex. These two cues (ITD and ILD) together enable the auditory system to localize a sound source within the external environment, calculating both the azimuth and distance from the observer.

Recently, robotics research has become interested in the ability to localize sound sources within the environment [3-4]. Audition is a vital sense for interpreting the world around us as audition enables us to perceive any object with an acoustic element. For localization and navigation purposes, the primary modality in robotics has been that of vision [5-6]. However, audition has some advantages over vision in that for us to visually see an object it must be within line of sight, i.e. not hidden by other objects. Acoustic objects however do not have to be within line of sight of the observer and can be detected around corners and when obscured by other objects.

This paper describes an acoustic tracking robotic system that is capable of sound source angle estimation and prediction along the horizontal plane. This system draws from some basic principles that exist in its biological equivalent, i.e. that of ITD, trajectory predictions, and the mammalian auditory cortex. Our system also has the ability to detect the angle of incidence of a sound source based on Interaural time difference.

2. Motivation for Research

How does the mammalian auditory system track sound sources so accurately within the environment? What mechanisms exist within the AC to enable acoustic localization and how can these be modeled? The AC of the mammalian brain works with excellent accuracy [7] and quick response times to the tracking and localization of dynamic sound sources.

With the increasing use of robots in areas such as service and danger scenarios [4], we are looking into the mechanisms that govern the tracking and azimuth estimation and predictions of sound sources within the mammalian AC to guide a model for sound source tracking within a robotic system. Our motivation comes from being able to create an acoustic sound source tracking robot capable of tracking azimuthally the angle of a dynamically moving stimulus.

With the scenario of interactive service robots, we can envisage the robot as a waiter in a restaurant serving drinks to the patrons. In order for this to be possible the customers would need to be able to attract the robot waiter's attention. The robot would need to detect the direction the sound comes from and attend to it. Fig. 1 shows an example of this scenario.

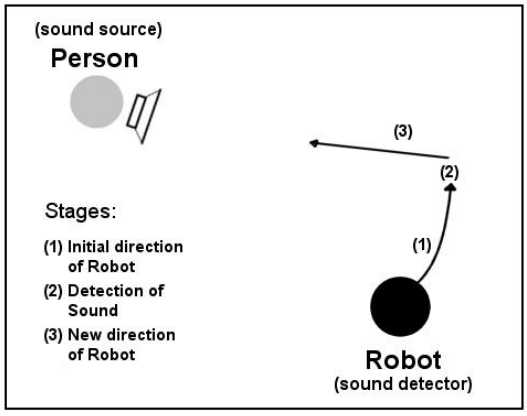


Fig. 1. Shows an example of a robot attending to sounds

3. The System Model

The model for our system has two main components; these are Azimuth Estimation and Neural Prediction. The first component in our model determines the azimuth of the sound source from the environment using Cross-Correlation and presents this angle to the Neural Predictor for estimation of the next predicted angle in the sequence using an RNN. The Neural Predictor receives a set of input angles passed from estimations of the angle of incidence from the azimuth estimation stage in the model and uses these to predict the angle for the robot to attend to next. The idea behind this is to enable the robot to move to the next position where it expects to hear the sound and then waits to see if it hears it. The system therefore has a faster response time to its tracking ability as the robot is not constantly calculating the position of the sound source, then attending and repeating this phase recursively, as this would mean the robot would be in constant lag of the actual position for the sound source. Instead our system has an active model for predicting the location of the sound source.

The system requires two valid sound inputs (as discussed in section 3.2). When the network receives its second input at time t_2 the network provided an output activation to attend to next. This is when the robot is informed to go at time t_2 as opposed to the passion the sound was detected at during time t_2 itself. Our system therefore provides a faster response in attending to the position of the dynamic sound source enabling more real-time tracking.

3.1 Azimuth Estimation

Azimuth estimation is the first stage of the overall system model and is used to determine the angle of incidence of the dynamic sound source from the environment. The azimuth estimation is performed by a signal processing variation of Cross-Correlation (Eq. 1). It has also been shown that the AC employs the use of Cross-Correlation as discussed by Licklider [8] for angle estimation. Therefore, we have employed Cross-Correlation to analyze the two signals *g(t)* and *h(t)* received at the left and right microphones in our system. Ultimately, Cross-Correlation as discussed in [9] is used for determining the ITD with the use of coincidence detectors [1].

Within our model the Cross-Correlation method is used to check *g(t)* and *h(t)* for the position of maximum similarity between the two signals, which results in the creation of a product vector $\boldsymbol{C}$ where each location represents the products of signals *g(t)* and *h(t)* at each time step. The robot records a 20ms sample of sound at each microphone resulting in an N x M matrix of 2 x 8820 where each row represents the signal received at each of the microphones. To correlate the two signals they are initially offset by their maximum length. At each time step signal *h(t)* is 'slid' across signal *g(t)* and the product of the signals is calculated and stored in the product vector $\boldsymbol{C}$.

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

Fig. 2 below shows an example of how the two signals *g(t)* and *h(t)* are checked for similarity. As can be seen in the graph of Fig. 2a we can see that the function starts

by offsetting the right channel *h(t)* to the beginning of the left channel *g(t)* and gradually 'slides' across until *h(t)* leads *g(t)* by the length of the matrix (graph in Fig. 2c). When the signals are in phase (shown by the shaded area in the graph of Fig. 2b) the resultant correlation vector will produce a maximum value at this time step position in the product vector ***C***.

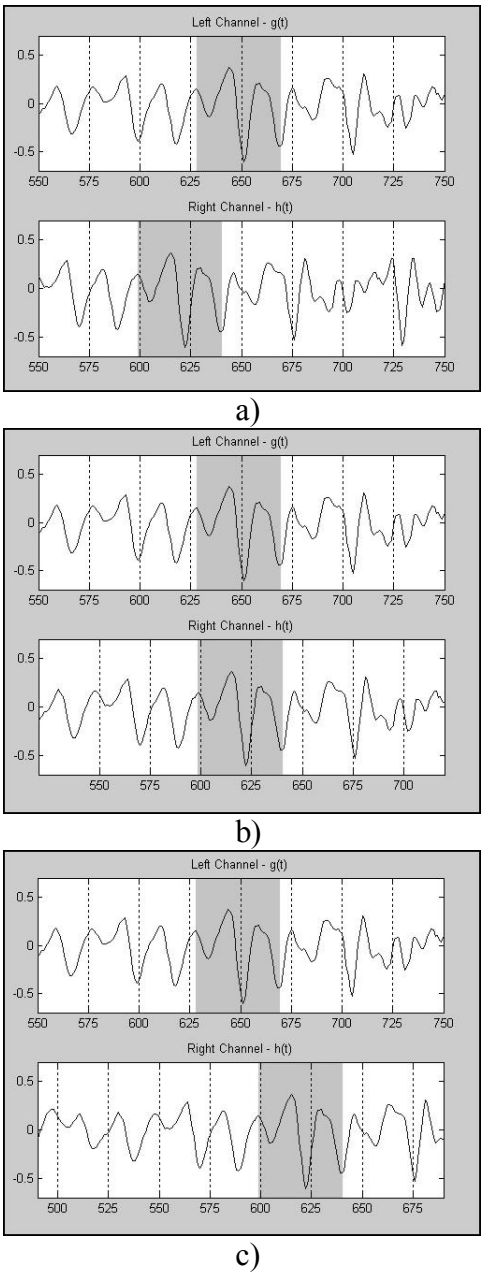


Fig. 2. Shows the 'sliding' of the signals presented to the cross-correlation function.

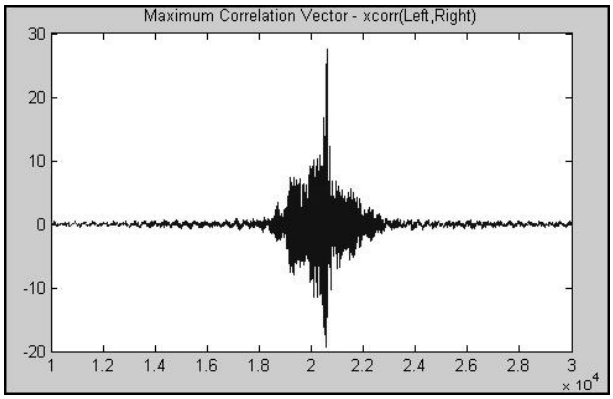


Fig. 3. Shows the product vector ***C*** of the Cross-Correlation of signals *g(t)* and *h(t)*

The maximum position within the resultant correlation vector represents the point of maximum similarity between *g(t)* and *h(t)*. If the angle of incidence was at 0^o then this would result in *g(t)* and *h(t)* being in phase with each other therefore resulting in the maximum value in ***C*** being in the middle location of the vector. See Fig. 4 for an example of the creation of a correlation vector from two slightly offset signals.

Therefore, to determine the amount of delay offsets for the ITD we subtract the location of the maximum point in ***C*** from the size of ***C***/2. We divide the correlation vector ***C*** by 2 as the method of cross-correlation creates a vector that is twice the size of the original signal (due to the sliding window) and therefore to find the mid point of the vector (i.e. zero degrees) we divide by 2. This result is used to help determine the angle of incidence of the sound source. The sounds within the system are recorded with a sample rate of 44.1 KHz resulting in a time increment Δ of 22.67μs. Therefore, each position within the correlation vector is equal to Δ and represents 2.267^{-5} seconds of time. Knowing the amount of time per delay increment and knowing the number of delay increments (from the correlation vector ***C***) then using equation 2 we can calculate the ITD, or more precisely in terms of our model, the time delay of arrival (TDOA) of the sound source between the two microphones.

$$\text{TODA} = ((\text{length}(\boldsymbol{C}) / 2) - \boldsymbol{C}_{\text{MAX}})*\Delta \tag{2}$$

This result gives us the time difference of the reception of the signal at the left and right microphones; this in turn is used in conjunction with trigometric functions to provide us with Θ the angle of incidence of the sound source in question. Looking at Fig. 5 we can determine which trigometric function we need to use to determine the angle of incidence.

We have a constant value for side 'c' set at 0.30 meters (the distance on the robot between the two microphones, see Fig. 10) and 'a' can be determined from the ITD or TDOA from Eq. 2 substituted into Eq. 3. Therefore, in order to determine the azimuth of the sound source we can use the inverse sine rule as shown in Eq 5.

$$Distance = Speed \times Time = 384 \times TDOA \tag{3}$$

where, Speed is the speed of sound = 384m/s at room temperature of 24°C at sea level. TDOA is the value returned from Eq. 2.

Signals: g(t) 111112222111 h(t) 111111112222

	Product Vector ***C***	
	Location	Value
1111122221110000000000	1	1
0000000000111111112222		
111112222110000000000	2	2
000000000111111112222		
11111222211000000000	3	3
00000000111111112222		
1111122221100000000	4	5
0000000111111112222		
111112222110000000	5	7
000000111111112222		
11111222211000000	6	9
00000111111112222		
1111122221100000	7	11
0000111111112222		
111112222110000	8	12
000111111112222		

Fig. 4. Shows how the sliding-window of the Cross-Correlation method builds the Correlation vector ***C***. As the signals get more in phase the value in ***C*** increases.

$$Sin\Theta = \frac{a}{c}, Cos\Theta = \frac{b}{c}, Tan\Theta = \frac{c}{b} \tag{4}$$

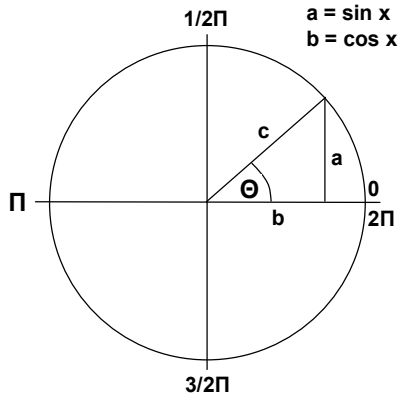


Fig. 5. Geometric diagram of sin(x) and cos(x)

$$\Theta = Sin^{-1}\frac{a}{c} \tag{5}$$

From these equations we can see that depending on the value of the TDOA we can determine the azimuth of the dynamic sound source. TDOA values can range from -90° to +90°. The values of Θ returned are used to provide input to the recurrent neural network of the second stage within the system for prediction of azimuth positions. This initial part of the model has shown that it is indeed possible to create a robotic system capable of modeling ITD to emulate similar robotic tasks. That is, we have looked at functional mechanisms within the AC and represented the output of these mechanisms within our robot model.

3.2. Speed Estimation and Prediction

Speed estimation and prediction is the second stage in our model and is used to estimate the speed of the sound source and predict the next expected location. It has been shown that within the brain there exists a type of short term memory that is used for such prediction tasks and in order to predict the trajectory of an object it is required that previous positions are remembered [10] to create temporal sequences.

Within this stage of our model, we create a recurrent neural network (RNN) with the aim to train this network to detect the speed of a sound source and provide estimated prediction positions for the robot to attend to. This stage in the model receives its input from the previous azimuth estimation stage as activation on the relevant neuron within the input layer of the network. Each neuron within the input and output layers represent 2° of azimuth, therefore an angle of 1° will cause activation on the first input neuron whilst an angle of 3° will cause activation on the second input neuron. As can be seen in Fig. 6 the input and out layers of the network have 45 units each with each unit representing 2° of azimuth. Therefore, the layers only represent a maximum of 90° azimuth, however as the sign (i.e. + or – angle recorded by the cross-correlation function) is used to determine if the source is left or right of the robots center then the network can be used to represent +90° and -90° thus covering the front hemisphere of the robot.

The RNN consists of four separate layers with weight projections connecting neurons between layers. The architecture of the network is as follows:

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

Fig. 6 shows the layout of the architecture within the network along with the fully connected layers. The network developed is based on the Elman network [11] which provides a method for retaining context between successive input patterns. In order for the network to adapt to sequential temporal patterns a context layer is used. To provide context the hidden layer has one-to-one projections to the context layer within the network (both the context and hidden layers must contain the same amount of

neurons). The hidden layer activation at time t_{-1} is copied to the context layer so that the activation is available to the network at time step t during the presentation of the second pattern within the temporal sequence. This therefore enables the network to learn temporal sequences which are required for the speed prediction task.

The input to the RNN is provided as activation on the input neuron that corresponds to the current angle calculated from the first stage in the model. In order for the network to make a prediction it must receive two sequential input activation patterns at times t_{-1} and t_0. This enables the RNN to recognize the temporal pattern and provide the relevant output activation.

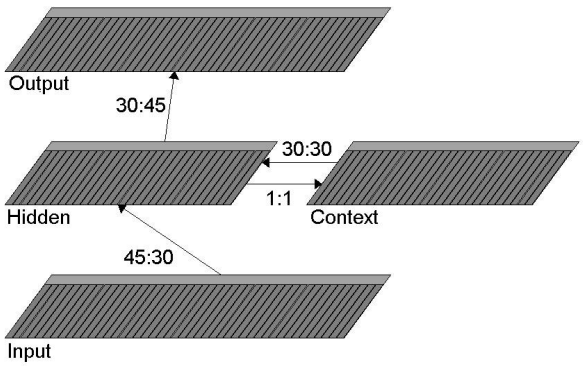


Fig. 6. Recurrent Neural Network architecture used in model.

Due to the RNN output activation only being set for the predicted angle to attend to, it is necessary to still provide the system with an angle of incidence to move to before the prediction is made. The angle recorded by the initial stage of the system is referred to as the perceived angle due to this being the angle of the sound source relative to the robot. The system also maintains a variable containing the current angle of the robot from its initial starting position of 0^o. Therefore the input activation for the RNN is calculated from Current angle + Perceived angle with the output activation being the angle from the starting position. Therefore, the angle to move to is RNN output angle – current angle.

The weight update algorithm for the network is based on that of the normal back-propagation as shown in Eq 7, with the stopping criterion for the network being set to 0.04 for the Sum Squared Error (SSE). This value was chosen as it was the highest value of the SSE that classified all patterns within the least number of epochs. The value of the SSE is checked after each epoch. If the change in the SSE is below 0.04 between epochs then training stops and the network is said to have converged, otherwise the weights are adapted and presentation of training patterns continues.

$$a_i^C(t+1) = a_i^H(t) \tag{6}$$

$$\Delta w_{ij}(n+1) = \eta \delta_j o_i + \alpha \Delta w_{ij}(n) \tag{7}$$

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

The RNN architecture provides the system with the ability to recognize a predetermined number of speeds (provided during training) from a sequential temporal pattern and therefore introducing a form of short-term memory into the model. After the RNN receives two time steps a prediction of the next location of the sound source is provided for the robot to attend to. This enables a faster response from the system and therefore enables a more real-time implementation of the sound source tracking system. This is due to the fact that the system does not have to wait for a subsequent third sound sample in order to determine the location in azimuth of the sound source.

3.3 Training the RNN

In order to train the RNN to recognize the various speeds, a separate training sub-group was created within the training environment for each individual speed. Each sub-group within the training environment contains the events required to train the network to individual speeds.

The events (input sequences i.e. angle representations) are activations of '1' on the neuron they represent and are presented for training to the network sequentially in the order expected from a sound source (and shown in Table 2). This is to ensure the network learns the correct temporal sequence for the speeds it needs to recognize and provide prediction for.

The environment shown in Fig. 7 presents the nine events in a sequential manner, that is, every time the pattern is presented to the network the events are given in the same order Event_0 → Event_8. However, the last two events (Event_7 and Event_8) within the training sub-group deviate from the temporal order and output activation of the first 7 events. These two events are provided to ensure the network does not only learn to provide activation output on the presentation of input activation on neuron 2 in Fig. 7 but also ensures that past context is taken into account and output activation is only set if a valid temporal pattern is provided to the network, in this case, at time t_{-1} activation on input neuron 1 and at time t_0 activation on input neuron 2 resulting in an output activation at time t_0 on the third output neuron in the network.

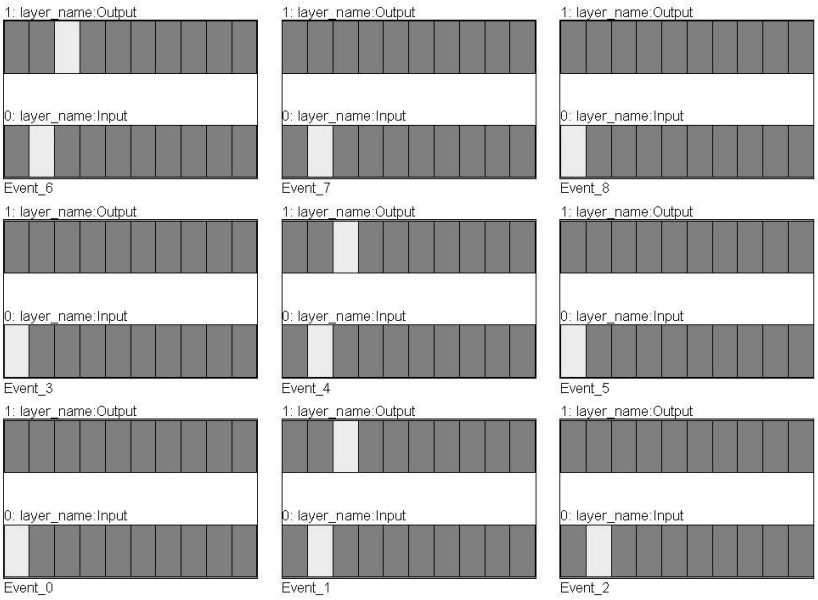


Fig. 7. Sub-group training environment for speed = 1 showing the required input activations in order to create an output activation remembering that the patterns are temporal.

Within the training environment 20 sub-groups were created with each sub-group representing a set speed, as each sub-group contains 9 training events this gives us a total of 180 events to present to the network. As previously mentioned, we trained the network by presenting the events within the sub-groups in a sequential order. However, each sub-group was presented in a random fashion to the network so as to prevent the network learning the presentation sequence of the sub-groups themselves. The network took on average 35000 epochs to converge, this varied slightly however due to varying initialization weights in the network when training.

4. Testing the System Model

The testing is carried out in two separate phases for the system model, with the azimuth estimation stage first being tested to ensure correct operation, i.e. correct estimation of the sound source along the horizontal plane. Once this stage is confirmed to be operating correctly, the output results are stored to file and used as the input to the second stage in that model, the Neural Predictor stage.

The results from the second stage on the model are checked against results from both predetermined input activation and randomly generated output activations to ensure the system does not respond erroneously due to unexpected input sequences, or incorrect weight updating and convergence.

4.1. Stage 1

We test the azimuth estimation stage of the model to ensure the correct azimuth values were being calculated and presented to the RNN. For this the robot was placed in the middle of a room with a speaker placed 1.5 meters from the center of the robot. The speaker was placed at 10 separate angles around the front 180^o of the robot. Each angle was tested five times with the results shown in table 1.

Table 1. Tests of Azimuth Estimation stage of model

Test	Actual Angle	Robot Position (Average)	Accuracy %
Test 1	-90	±4	95.5
Test 2	-50	±2	96
Test 3	-40	±1	97.5
Test 4	-30	±0	100
Test 5	0	±2	98
Test 6	+10	±2	80
Test 7	+20	±1	95
Test 8	+35	±2	94.3
Test 9	+45	±2	95.6
Test 10	+70	±3	95.7

As can be seen from table 1 the maximum average error was ±1.9^o. That is, the averages in column 3 summed and divided by number of test cases to give the average system error (±19/10 = 1.9). As shown in [7] the human auditory system can achieve an accuracy of ±1.5^o azimuth. Therefore, the results from the initial tests for this stage in our model show that the use of cross-correlation for calculating TDOA and ultimately the angle of incidence is an effective system for determining the azimuth position of a sound source.

Furthermore, the results passed into the second stage of our model are also accurately representative of the actual position of the sound source within the environment and therefore a useful input into the RNN for predicting the next angle.

4.2 Stage 2

Testing of the RNN after training was done with the aid of azimuth results, i.e. a data file was created with the angles returned of the initial stage of the model as the sound source was moved around the robot at various speed levels. This data was then presented to the network in order to check the response of the system to actual external data as opposed to simulated environments.

Fig. 8 shows the response of the network when the test data presented activation in an accepted sequential order. The first angle presented in Fig. 8a was within the range 0^o → 2^o and therefore provided input activation to the first neuron. Next, in Fig. 8b the angle presented was within the range 2.01^o → 4^o and therefore activated input neuron 2; this resulted in a recognizable temporal pattern therefore providing output activation for the next predicted position as shown in the output layer of the network in Fig. 8b.

Fig. 9 shows the response of the RNN to a different sequence (speed) to that presented in Fig. 8. Fig 9a shows the first pattern at t_{-1} with an activation on the first input neuron, representing an azimuth estimation of $0^o \rightarrow 2^o$. The second pattern presented at time t_0 (Fig. 9b) is on the 11 input neuron and so represents an azimuth angle of $20^o \rightarrow 21.9^o$. Output activation is also provided in Fig. 9b on the 21st output neuron representing an angle of azimuth of $40^o \rightarrow 41.9^o$ for the robot to attend to.

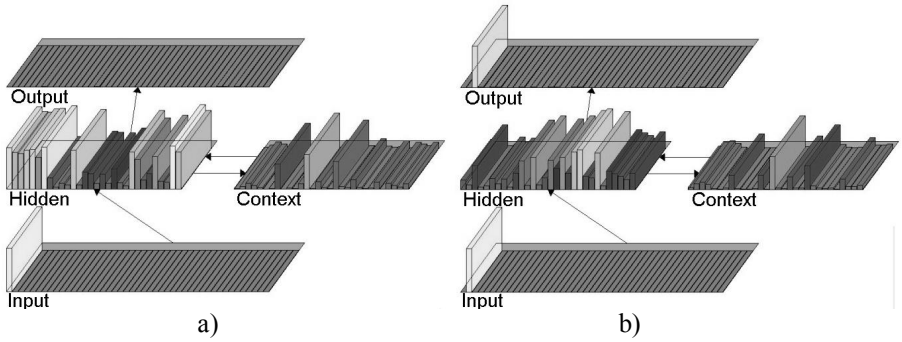


Fig. 8. Shows the response of the RNN after input activations at t_{-1} and t_0 for speed 1

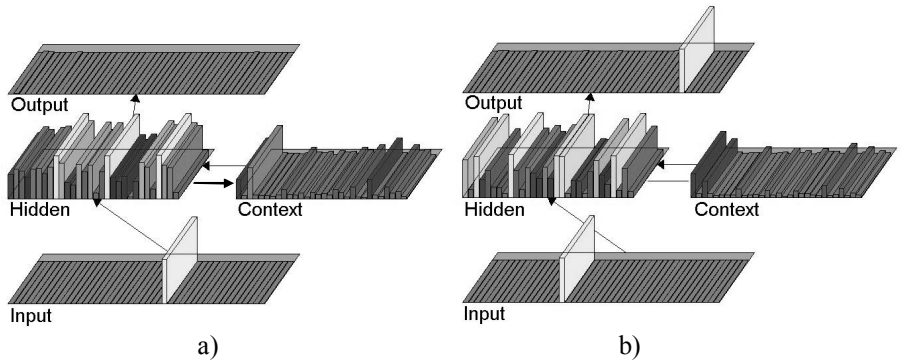


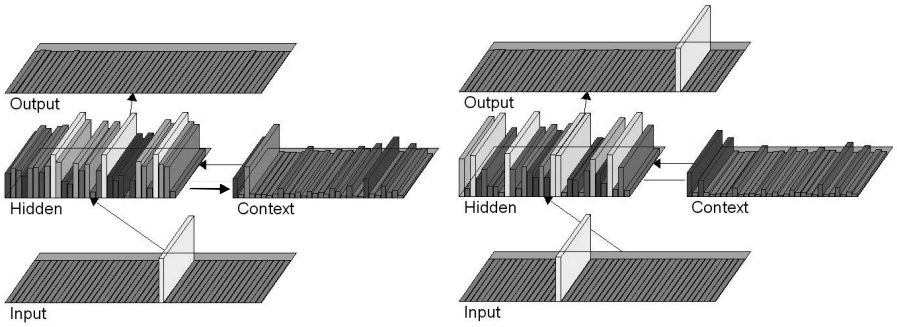
Fig. 9. Shows the response of the RNN after input activations at t_{-1} and t_0 for speed 8

There are always going to be cases when a set of angles presented to the network do not match any predetermined speed representations. To ensure the system provides correct output activation for unforeseen input sequences a testing environment of 10000 randomly generated events was created.

Once the environment was created it was first passed through an algorithm to analyze the input layer activations within the training environment to determine the correct output activation that should be seen; these 'desired' input (with calculated) output activation patterns are then stored to file to later compare with the actual output activation received from the network once the randomly generated test environment has been passed to the system.

The output activations of the network were recorded and compared with the 'desired' stored results to ensure they matched. The comparison showed that from the randomly created test environment only one pair of unforeseen sequences caused a

misclassification. Fig. 10 shows the particular misclassification found within the RNN during the specific temporal pair of input sequence patterns. Fig. 10a shows at t_{-1} input activation falls on neuron 28 and at time t_0 Fig. 10b shows that input activation falls on neuron 18. Clearly this is not one of out trained speeds (as the sequence goes backwards) however output activation is set at time t_0 to neuron 39.



a) b)

Fig. 10. Shows an misclassification within the trained RNN providing undesired output activation to the system.

Table 2. Representation of input activations for the first 4 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……..

Table 2 gives an example of the input sequences for the first four trained speeds. The binary pattern represents activation on the input neurons of the RNN where '1' shows activation on the relevant neuron. As can be seen from the table, each speed is determined by the increment in neuron activation between t_1 and t_2 in the temporal sequence.

The results from the testing of the RNN shows that it is possible to encode several temporal patterns within a RNN using a context layer to act as a form of short-term memory within the system to provide history information for the network to use in classifying temporally encoded patterns. With the exception of the temporal sequence shown in Fig. 10 which shows a misclassification in the system for a temporal sequence pair, all other combinations of sequences within the testing environments 10000 events provided correct desired output activation i.e. either no output at all for undesired temporal pairs or single neuron output activation for desired temporal sequence pairs.

With the introduction of new sub-groups within the training environment it is possible to remove anomalies from the network. This would be accomplished by including the temporal sequence shown in Fig. 10 but having no output activation set. However misclassifications will not be detected until the sequence that generates them is presented to the network.

5. Discussion

Much research has been conducted in the field of acoustic robotics. However, many of the systems developed have concentrated more on the principles of engineering rather than that of drawing inspiration from biology. Auditory robots have been created which use arrays of microphones to calculate independent TDOA between microphone pairs [12]. Research has also been conducted into the AC of a robotic barn owl [13] for attending to objects within the environment. However, such systems include other modalities such as vision to aid the localization of the spatial object.

The currently developed model described here, whilst providing excellent results for sound source azimuth estimation and tracking can not adapt in an intuitive way to the dynamics of real world acoustic objects. Further study is currently being conducted into creating an architecture that can learn to recognize the temporal sequences of new speeds the system may encounter but does not have in its training set. Using this adaptive network to recognize new temporal patterns, it may also be possible for the network to learn how to recognize acceleration and deceleration patterns through this adaptive model. This adaptive network would provide the system with the ability to more accurately track sound sources whose dynamic motion is not a fixed constant but rather varies its speed randomly.

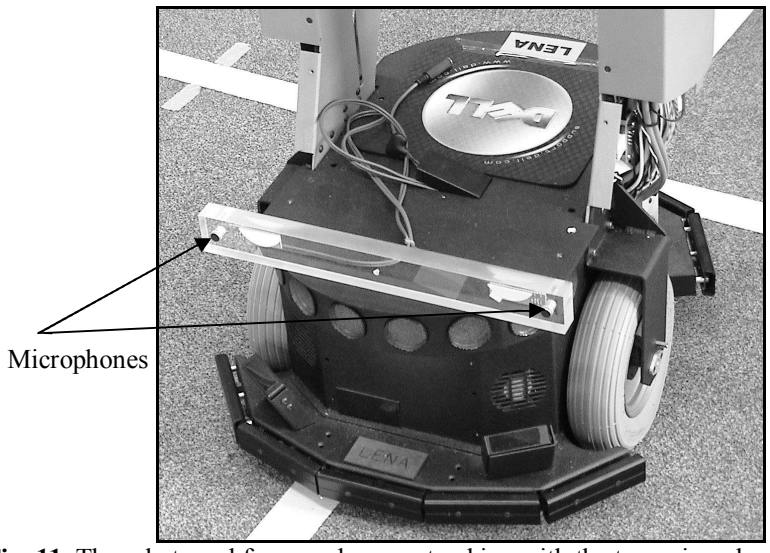


Fig. 11. The robot used for sound source tracking with the two microphones as ears.

6. Conclusion

A hybrid architecture has been presented with inspiration drawn from the mechanisms that have been shown to exist within the AC [1, 7, 14] of the mammalian. By using biological inspiration we can take advantage of the cues and mechanisms that already exist to build our model. The model has been shown to utilize the ITD cue to determine the angle of incidence of the sound source and present this to a RNN for temporal processing to determine the current speed and predict the next location for the robot to attend to. The hybrid architecture shown has proven to have excellent potential for developing robotic sound source tracking system which draws inspiration from their biological counterpart. The results of our model have shown comparable with the capabilities of the human AC with the azimuth localization differing by an average of $\pm 0.4^{o}$.

As more information on the workings of the AC becomes known it would be possible to further adapt and create neural network architectures that emulate the functionality of the various components of the AC giving rise to robotic system which operate in the acoustic modality in much the same manner as the mammalian.

References

[1] Joris, P.X., Smith, P.H., and Yin, T.C.T., Coincidence detection in the Auditory System: 50 years after Jeffress. Neuron, 1998. 21(6): p. 1235-1238.

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

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

[4] Macera, J.C., et al., *Remote-neocortex control of robotic search and threat identification.* Robotics and Autonomous Systems. 2004. 46(2): p. 97-110.

[5] Bohme, 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.

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

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

[8] Licklider, J.C.R., *Three auditory theories, in Koch ES (ed) Psychology: A Study of a Science*. Study 1, Vol. 1, New York: McGraw-Hill: p. 41-144.

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

[10] Kolen, J.F., Kremer, S.C., The search for context, in A Field Guide to Dynamical Recurrent Networks, 2001. IEEE Press. p. 15-17.

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

[12] Valian, J-M., et al. *Robust Sound Source Localization Using a Microphone Array on a Mobile Robot.* In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robotics and Systems, p. 1228 – 1233

[13] Rucci, M., Wray, J., Edelman, G.M., *Robust localization of auditory and visual targets in a robotic barn owl.* Robotics and Autonomous Systems 30 (2000) 181-193.

[14] Griffiths, T.D., Warren, J.D., *The planum temporale as a computational hub.* TRENDS in Neurosciences. 2002, 25(7): p. 348-353.