A Recurrent Neural Network for Sound-Source Motion Tracking and Prediction

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

Abstract – Recurrent neural networks (RNN) have been used in many applications for both pattern detection and prediction. This paper shows the use of RNN's as a speed classifier and predictor for a robotic sound source tracking system. The system requires extensive training to classify all possible speeds to enable dynamic tracking of the most prominent sound within the environment.

Keywords: Tracking, Recurrent Neural Network, prediction, robotics, cross-correlation.

I. INTRODUCTION

Our ability to interact within our external environment using the various senses available to us has been of interest to people for many years, both in research to further understand the workings of the human system (neurological and physiological) [1-2] and for manufacturers, for the development of robotic systems. Many researchers have looked into creating systems that are able to interact on social levels as well be able to navigate within the environment [3]. This paper describes a robotic system capable of sound source localization and tracking with prediction capabilities. The motivation for this research is taken from the biological system which is able to accurately and speedily track multiple sounds of interest with respect to background noise.

II. BACKGROUND

Tour guide robots are one example of such 'sociable' robotic systems [4]. These tour guides move around their environment whilst avoiding obstacles they may encounter whilst also being able to interact with the people on a seemingly intelligent level by answering any questions that may be posed from the audience. Sound source localization and tracking is an important task for such robots in order to improve their speech understanding capabilities. For example due to the formation of the human pinna (ears) when trying to improve the signal of a speaker, we face the direction of the sound, this helps by improving the signal to noise ratio (SNR) by attenuating the surrounding sounds. This is also important as if we are surrounded by several sound sources we wish to only focus on the particular sound of interest and therefore reduce the signals from undesired sources. Our aim is to develop the system with inspiration taken from that of the mammalian auditory system as mammals are extremely efficient in the way in which they localize sound sources with some animals reaching an accuracy of $\pm 1^{o}$ on the horizontal plane and $\pm 5^{o}$ with respect to elevation [5].

III. MODEL

The system consists of three separate stages in order to have the ability to track sound sources azimuthally, the first of which is the cross-correlation system. The cross-correlation part of the system is used to estimate the sound source position [6].

The second stage within the system enables us to determine to direction of motion of the sound source, i.e. to the left or right of the robots center position.

Finally the third stage is the RNN neural processing stage used for the prediction of the dynamically moving sound source within the environment.

A. Cross-Correlation

This stage calculates the angle of incidence of the detected sound source which is subsequently passed to the neural processing stage enabling a prediction of the next location along the azimuth plane to be made.

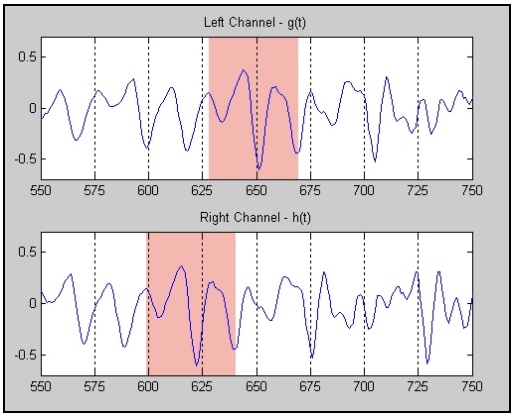

Figure 1. Analysing two signals to determine azimuth

Figure 1 shows the cross-correlation 'window' of the two signals received at the left and right microphones during stage one of the system. Cross-correlation 'slides' the two signals *g(t)* and *h(t)* across each other and results in the creation of a product vector at each time sample step t, the maximum position of the correlation vector ***C*** therefore represents the position of maximum similarity or correlation. The shaded area seen in Figure 1 shows the parts of the two signals *g(t)* and *h(t)* which are the same sound source only lagged from each other due to the separation of the left and right microphones.

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

Equation 1 shows the formula used to compute the cross-correlation of the two signals *g(t)* and *h(t)*. This creates a correlation vector which is analyzed to ultimately calculate the angle of incidence.

B. *Motion direction*

The second stage in the system is used for determining the direction of motion, i.e. whether the object is traveling left to right, or right to left thus allowing the robot to know in which direction to move. This is achieved by analyzing the correlation vector ***C*** which results from stage 1 above. Depending on the position of the sound source (i.e. to the left or right of the center of the robot) will determine whether the maximum correlation position within ***C*** is to the left or right of the center of the vector itself. Thus, this result is used to set the direction of the motors for the movement of the robot. Using this method reduces the need to for the network to process the left and right sides of the robot, taking into account the motion direction.

C. *Neural Processing*

This is the final stage of the system (and the main focus of this paper) which consists of the neural architecture for aiding in the prediction of the next location along the azimuth trajectory of the sound source.

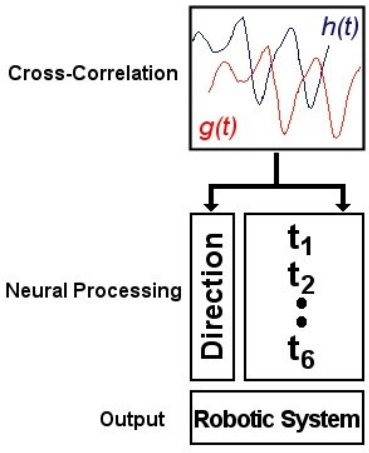

Figure 2. Overview of System Architecture

Figure 2 shows how the three separate stages of the system are connected and ultimately their order of execution within the time frame of the tracking of sound sources.

IV. NEURAL ARCHITECTURE

The neural architecture of the system has been developed using the application tool PDP++ [7] which is a research tool for the creation and development of many different types of recurrent neural networks. PDP++ includes the learning algorithms and connection specifications for most types of existing networks. As well as the ability to create custom specs. The recurrent neural network which was developed for our sound source tracker consists of four layers, they are:

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

The auditory localization of the system is capable of detecting sounds within 0^o – 90^o left or right of the centre of the system and an accuracy of $\pm 1.5^o$. Initial experiments were conducted with 20 input neurons; however this introduced a large variable error into the system as each neuron represented 4.5^o of the environment space, therefore increasing the number of neurons to 45 reduced the possible error by 2.5^o to 2^o per neuron.

The recurrent neural network system is constructed using the standard feed-forward back propagation weight updating. However, to predict the next location within the trajectory the system needs to learn temporal patterns and so requires a form of short-term memory provided by the context layer to enable the prediction tasks [8] as is also evident within the human auditory cortex [5].

The hidden layer provides one-to-one projections to the context layer as shown in figure 3. This enables the activation of the hidden layer at t_{-1} to be copied across and be available to the network at time t_0. Figure 3 shows the layout of the network in terms of number of neurons and projection direction.

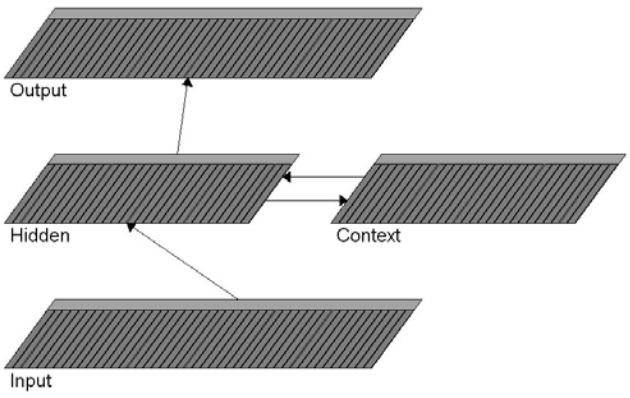

Figure 3. Layout of recurrent neural network architecture used

This architecture is that of an Elman recurrent network [9], where the hidden unit activations at time t_{-1} are used to provide the context for the system.

The hidden layer provides this context by copying to the context layer via one-to-one projections whose weights are set to 1.0 so as to provide no negative or positive bias. However this particular architecture developed varies from the Elman network in its weight algorithms. Here we have adopted the standard back propagation weight update algorithm:

$$\Delta w_{ij} = \eta \delta_i a_j \qquad (2)$$

where, $\eta = 0.25$

As mentioned the one-to-one projection prevents the context unit activations being altered during copying and therefore they remain unchanged at t_0 and are the same as the hidden layer activations at t_{-1}:

$$a_i^C(t+1) = a_i^H(t) \qquad (3)$$

V. The Methodology

When the system first detects the sound source at time t_1 the cross-correlation method of stage 1 (above) calculates the angle of incidence relative to the robots frame of reference. The angle measurement is propagated to the neural stage of the system as the first input in the temporal pattern for the network to recognize and therefore ultimately predict the third location of the trajectory of the dynamic sound source and allowing the robotic system to 'meet' the acoustic object there. As the network requires two angles before it can predict a third it is trained to only give an output when it has received two angle positions from the cross-correlation stage. Only when angles at time t_1 and t_2 have been presented to the network will their be activation on the output layer of the network and thus providing movement instructions to the robot and allowing the system to move to the predicted location of the sound source at time t_3.

VI. Environments

The system training environments used in the network have been set up to ensure the network will only respond by providing output activation after two consecutive time steps with valid patterns that are defined in the training sets, an example of the training sets are shown in table 1. This therefore ensures the system remains temporal in its learning and prediction tasks. That is, that only valid 'sequential temporal' patterns will provide output activation as opposed to only one angle providing such output. For example, box 1 shows the desired time-series output for the system given two consecutive input patterns.

	IP OP
P_1	
t_1	100000 000000 <- no desired output
t_2	010000 001000 <- predicted pos
P_2	
t_1	100000 000000 <- no desired output
t_2	001000 000010 <- predicted pos

Box 1 – Shows an example of expected outputs for specific input pattern sequences.

As is shown in box 1, upon the first presentation of an angle the system does not provide a prediction as it is unaware of the speed the object may be traveling due to 'all' system speeds having the possibility of the same initial bit pattern and therefore unsure of the target location. Only after the presentation of the second angle can the system determine the pattern and provide a prediction for the next position of the object along its trajectory path.

VII. Training

The training method for the recurrent neural network used is that of the standard backpropagation through time [9] algorithm with the weights between the hidden layer and its context layer being one-to-one with a value of 1.0 (constant) as described by the Jordan/Elman network definitions [9].

TABLE I. Sample of bit patterns for first nine speeds.

Speed	Network input neurons
All t_1	10000000000000000000……..
1 t_2	01000000000000000000……..
1 t_n	00100000000000000000……..
2 t_2	00100000000000000000……..
2 t_n	00001000000000000000……..
3 t_2	00010000000000000000……..
3 t_n	00000010000000000000……..
4 t_2	00001000000000000000……..
4 t_n	00000000100000000000……..
5 t_2	00000100000000000000……..
5 t_n	00000000001000000000……..
6 t_2	00000010000000000000……..
6 t_n	00000000000010000000……..
7 t_2	00000001000000000000……..
7 t_n	00000000000000100000……..
8 t_2	00000000100000000000……..
8 t_n	00000000000000001000 ……..
9 t_2	00000000010000000000 ……..
9 t_n	00000000000000000010……..

To ensure the system learns the temporal states required each training pattern presented is stored as a sub-group within the training environment. These events (input

sequences) within each sub-group are presented to the network in sequence to maintain temporal order, whilst each sub-group is presented in a random order to ensure maximum fitting of the weights.

For this particular application the weights were updated after each sequence was presented to the network as opposed to updating after each sub-group, as recommended in [10] this prevented the system oscillating with regards to the weights, the system also converges much quicker.

Table 1 shows the bit pattern input for a sample of the first 9 possible speeds of motion the system was trained to recognize.

As the robots coordinates frame of reference is always head centered then each speed always begins with the same starting pattern (I.e. t_1 in table 1), therefore this reduced the amount of training patterns required for the system to learn as it was not required to train the network with all possible speeds at every possible input neuron.

Figure 4. Sample of training data for Speed 1

Fig 4 shows the sample training environment for training the network to recognize the first speed in table 1 (note however that for purposes of presenting the diagram in this paper only the first ten input and output neurons are shown on the events. However, this does not effect the information trying to be presented as only the first three neurons are of importance for the first speed). Subsequent patterns were presented in the same manor, with nine events for each sub-group giving a total set of events of:

$9_{events} \times 20_{sub\text{-}groups} = 180$ training events

VIII. Testing

Once the system was fully trained two testing environments were created, the first environment contained 1000 randomly generated events. This environment was used to determine the networks ability to operate correctly even with spurious data. For example, if the two temporal inputs where first to activate neuron eight on the input and then neuron six in that sequence the system should not provide an activation output as this would be an invalid sequence. The second environment used was a set of 200 manually created events. These events contained valid temporal sequence information and were used to check the response of the system to the desired input and output.

As discussed earlier in order to be able to predict the next possible location of the spatial object we need to be able to determine the speed at which the object is traveling, for this, two time increments are required, t_1 and t_2 for prediction of t_3. Figures 5 and 6 below show the first two time steps being presented to the network sequentially resulting in the desired output of the next position of the dynamic sound source determined by its current speed.

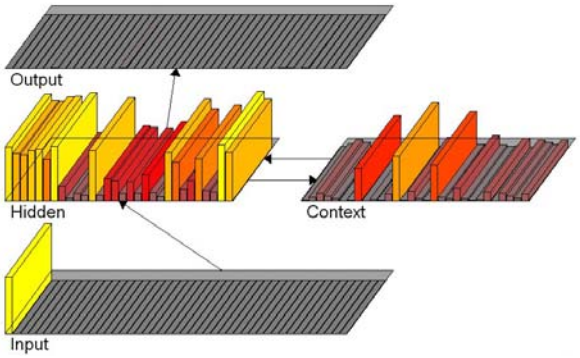

Figure 5. Input / Output of speed 1 at t_1

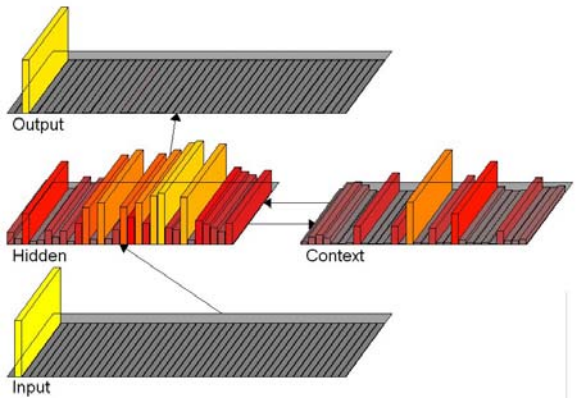

Figure 6. Input / Output of speed 1 at t_2

At t_1 the network presents the pattern 'All t_1' shown in table 1. As expected the output at this stage remains with all output neurons at zero activation, the pattern for speed 1 at time t_2 is next presented to the network and the output position for the robot to attend to is provided as activation on the output layer see Figure 6.

IX. Results

One of the required factors desired from the system is the ability for the network to provide the output within a restricted finite time increment so as to enable the robot to move to the position provided by output activation at time t_3

and be ready for the next input. The network developed for our system was remotely installed on a Pentium 4 3GHz PC and received input and provided output via the use of program sockets.

The two testing environments mentioned in section VIII were presented to the network and the output was recorded to file for analysis. The randomly generated environment gave results of 92% as certain 'random' combinations of undesired sequences gave anomalous results, Figure 7 and 8 shows one of the anomalies. However, when the manually created 'desired' environment was presented to the network results were 100% this could be due to the second environment not reproducing the particular pattern that caused the anomaly.

After 'stand alone' testing the network was connected to the robotic system. Each sound sample was 20ms in length and recorded at 1 second intervals. The system was able to actively track a dynamic sound source moving at a maximum speed of 34^{o} per second, this was due to the restriction in the maximum speed of the current robot being used combined with floor friction.

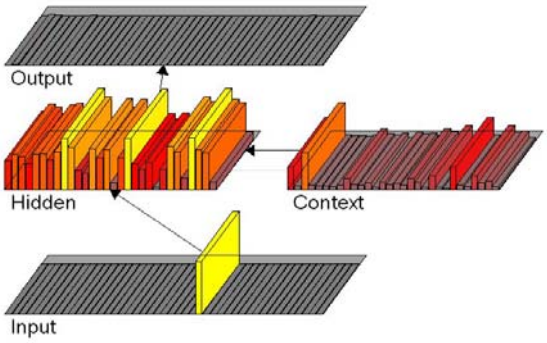

Figure 7. Test w/random environment T_1

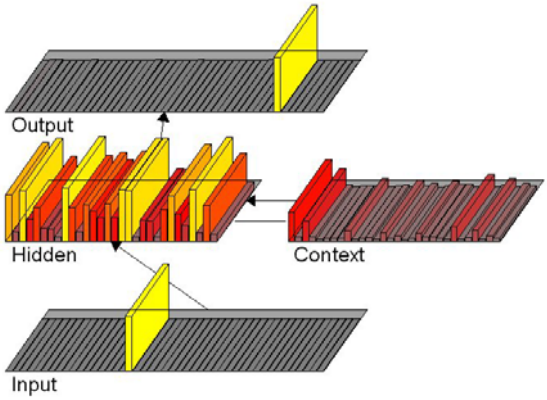

Figure 8. Test w/random environment T_2

X. Conclusions

The conclusions drawn from the results of these experiments have shown that a recurrent neural network is a viable and accurate method of predicting motion trajectory for a robotic sound source tracking system. It has been shown that the system is capable of learning spatio-temporal information and using this to predict the next location of the stimulus within the environment. The system was also able to adapt to varying speeds for the same source however it was not possible to calculate acceleration.

XI. Further Work

Work is now being conducted to enable the robotic tracking system to be able to learn on-line, that is, to be able to update its internal representation of the sound source within the environment therefore enabling a more 'real' representation of spatio-temporal motion within the environment in which it exists as well as a method for acceleration detection.

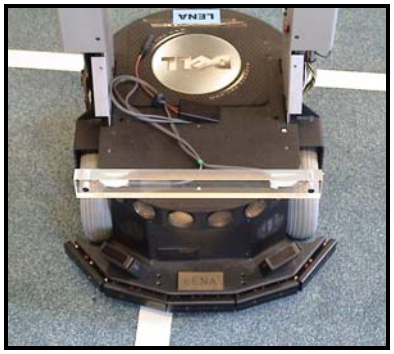

Figure 9. This shows the robot used for development of the system.

XII. References

[1] Cynthia Breazeal and Brian Scassellati. Robots that imitate humans. TRENDS in Cognitive Sciences, Vol. 6(11) 2002, pages 481-487

[2] Cynthia Breazeal. Towards Sociable Robots. Robotics and Autonomous Systems, Vol. 42(3-4) 2003, pages 167-175

[3] Hans-Joachim Böhme, Torsten Wilhelm, Jürgen Key, Carsten Schauer, Christof Schröter, Horst-Michael Groβ, Torsten Hempel. An approach to multi-modal human-machine interaction for intelligent service robots. Robotics and Autonomous Systems, Vol 44(1) 2003, pages 83-96.

[4] Illah R. Nourbakhsh, Judith Bobenage, Sebastien Grange, Ron Lutz, Roland Meyer and Alvaro Soto. An affective mobile robot educator with a full-time job. Artificial Intelligence, Volume 114, Issues 1-2, 1999, Pages 95-124

[5] Jens Blauert. Spatial Hearing – The Psychophysics of Human Sound Localization. 1997, Table 2.1, page 39.

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

[7] Randall C. O'Reilly PDP++ Neural Simulation http://www.cnbc.cmu.edu/Resources/PDP++//PDP++.html

[8] Robert Callan. The Essence of Neural Networks. 1999 ISBN: 0-13-908732-X pp 100-114.

[9] A Field Guide to Dynamical Recurrent Networks. John F. Kolen, Stefan C. Kremer. 2001 ISBN: 0-7803-5369-2 pp 18-23.

[10] Chadley K. Dawson, Randall C. O'Reilly, and James L. McClelland. The PDP++ Software Users Manual. http://www.cnbc.cmu.edu/Resources/PDP++/manual/pdp-user_toc.html Section 14.2