

Localizing moving underwater objects using an artificial lateral line and neural networks

Luuk H Boulogne, s2366681, lhboulogne@gmail.com,
Supervisor: Sietse M van Netten

Artificial Intelligence, University of Groningen

January 30, 2016

Abstract

With their lateral line, fish are able to sense water velocities. Using this organ, they can determine the location of nearby moving objects. In this research project neural networks are used to extract a source location from excitation patterns measured by fluid velocity sensors along a 1D lateral line. These sensor measurements are simulated using a theoretical model. The networks used are Echo State Networks and Multilayer Perceptrons, both with 16 and 32 input sensors. The parameter settings of the networks were not fully experimentally determined. The performance of the different networks for the chosen settings is compared. Their noise robustness with respect to the input excitation pattern amplitude divided by the standard deviation of the Gaussian distribution from which the noise was sampled is analyzed. This research shows that, for the chosen settings, Multilayer Perceptrons with 16 input sensors have the best performance on average, although the best performing individual network is an ESN with 16 input sensors with a mean error in Euclidean distance of less than 0.2 in a field of 1×2 . In an x,y -plane with the x -axis parallel to the lateral line, the y -coordinate is more accurately detected by networks with 32 sensors than by networks with 16 sensors, while the x -coordinate is best detected by networks with 16 sensors. The accuracy of all networks is decreased in the corners of the field. The noise robustness of both types of networks is similar.

1 Introduction

1.1 Research question

Along the sides of their body, fish have an organ that enables them to detect nearby moving underwater objects. This organ is called the lateral line. It is a mechanoreceptive system that allows for the detection of water flow (Coombs, Janssen, and Webb, 1988, pp. 554-555). The lateral line organ consists of an array of neuromasts, that can detect water velocities. Fish have many uses for their lateral line. It allows them to detect e.g. prey (Hoekstra and

Janssen, 1985) and facilitates schooling (Partridge and Pitcher, 1980), also when no light is available.

A vital component of an artificial lateral line is an efficient method that processes the detected signals. This paper focuses on setting the first steps in processing lateral line excitation patterns using neural networks. The network types used in this research are Multilayer Perceptrons (MLP) trained with the backpropagation algorithm (Rumelhart, Hinton, and Williams, 1985) and Echo State Networks (ESN) (Jaeger, 2002).

In previous research, theoretical models are de-

scribed that predict excitation patterns along a 1D lateral line, given a source on a 2D plane that is vibrating with a specific angle with respect to the lateral line (Ćurčić-Blake and van Netten, 2006) or moving in a specific direction (Franosch, Sichert, Suttner, and van Hemmen, 2005).

Using these models, excitation patterns for different locations and directions of a moving source can be computed. The related research question answered by this bachelor project is the following: Do neural networks have the ability to accurately decode the 2D location of a moving object from the excitation patterns along an artificial lateral line?

1.2 Significance of this research

Retrieving a moving source location from excitation patterns along an artificial lateral line is the first step in imaging the environment of such a lateral line. If this is possible, more advanced imaging of surroundings can be explored. Advanced versions of this technology might be able to image entire underwater environments.

The LAKHsMI EU H2020 Blue Growth 09 program has the goal of developing a technology that measures differential pressure and temperature underwater and uses these measurements for high resolution imaging of its surroundings (LAKHsMI, 2015). This technology is inspired by the biophysics of hydrodynamic sensing of fish. The aim of this bachelor project is to acquire knowledge that can aid this program in reaching its goal.

2 Methods

Since MATLAB was chosen as the standard computing language for the LAKHsMI project, MATLAB Student R2015a is used for the programming in this research. Also, the code produced for this research is anticipated to be used in future projects.

The optimal parameter settings of the networks have not fully been experimentally determined. The networks seemed to perform relatively well with the chosen parameters.

2.1 Echo State Network

2.1.1 Basic ESN

This section illustrates the basic properties of a typical ESN. A more extensive description about how the ESN was constructed and trained can be found in Jaeger (2002, pp. 29-32).

Figure 1 shows a schematic representation of an ESN. An ESN is a Recurrent Neural Network (RNN) that consists of an input layer, an output layer and a dynamic reservoir (DR) in between them (Jaeger, 2002, chapter 6). This reservoir consists of hidden nodes that are connected with each other and themselves through weights. The input nodes of the ESN are connected to the nodes of the DR, which causes the DR to contain a high dimensional representation of the input. Optionally, ESNs also have connections that go back to the DR from the output layer. The connections between nodes within the DR and the connections from the output layer to the DR make the ESN recurrent, causing it to have short term memory (Jaeger, 2002, chapter 7).

The weight matrix representing the internal weights of the DR must be constructed so that the spectral radius, which is the largest absolute eigenvalue of the matrix, is smaller than 1. If this is not done, the activation of the internal nodes snowballs, causing nonsensical output.

ESNs that have a DR with a spectral radius smaller than 1 have the Echo State Property. This means that activity of the DR dies out over time when it doesn't receive input anymore. The input of a large number of time steps back has little to no effect on the internal and output states of the ESN, while input of few time steps back has a significant effect. This gives the ESN short term memory.

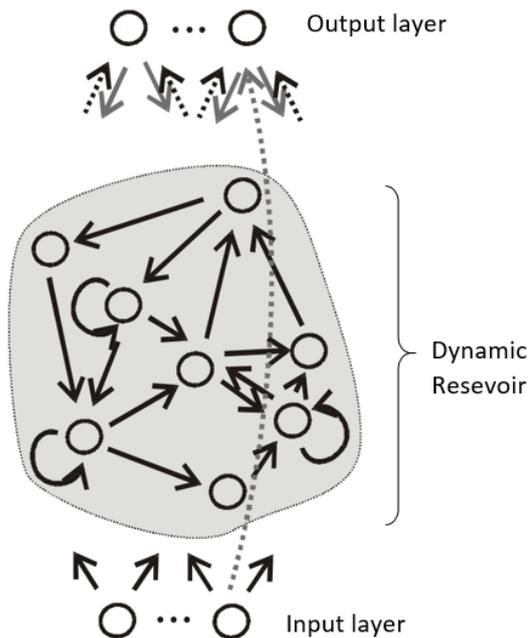


Figure 1: (Jaeger, 2002, pp. 7) Schematic representation of a basic ESN. Weights are represented by arrows. Grey weights are optional. Only dotted weights are trained.

Because of this, and because before the first few data points presented to the network, not many input patterns have been presented, the output of the first few time steps is often inaccurate. To wash out the effects of the initial network state, the network output of the first few samples is discarded during training.

During training, only the weights connecting the DR to the output layer are trained in the way described in Jaeger (2002, section 6.3). All other weights are initialized to random values. These values are not altered after initialization.

2.1.2 Why an ESN?

Among other things, the excitation pattern along a lateral line caused by a source depends on the location and movement direction of that source (Ćurčić-Blake and van Netten, 2006). When an excitation pattern is presented, an ESN has access to information about previous

excitation patterns, presented some time steps earlier, because of its short term memory. Since moving objects often do not change their direction of movement abruptly, the short term memory of ESNs was selected to decode the location of a moving source.

Furthermore, because only one layer of weights is trained, an ESN can be trained in short time.

2.1.3 Specifications

The ESN used in this research consists of an input layer of variable size (32 or 64 nodes), a DR of 40 nodes and an output layer that has two nodes to represent the 2D location of the source. The input layer is fully connected to the DR and the DR is fully connected to the output layer. The nodes in the DR are only sparsely connected in such a way that the DR only contains 10% of all possible connections. This causes the dynamics of the internal nodes to vary more. The input and output layers are not directly connected, but the output layer is fully connected back to the DR.

The weights from the input layer to the DR were initialized uniformly in the range $[-0.5, 0.5]$ and the weights from the output layer back to the DR were initialized uniformly in the range $[-0.25, 0.25]$. The weights in the DR were taken from a uniform distribution with a mean of zero so that the DR has a spectral radius of 0.8.

To calculate the activation of the nodes in the DR and the output layer, the identity function is used. A washout time of 50 time steps was used during training.

2.2 Multilayer Perceptron

An MLP is a feedforward network that consists of an input layer, one or more hidden layers and an output layer. Figure 2 shows an example of an MLP.

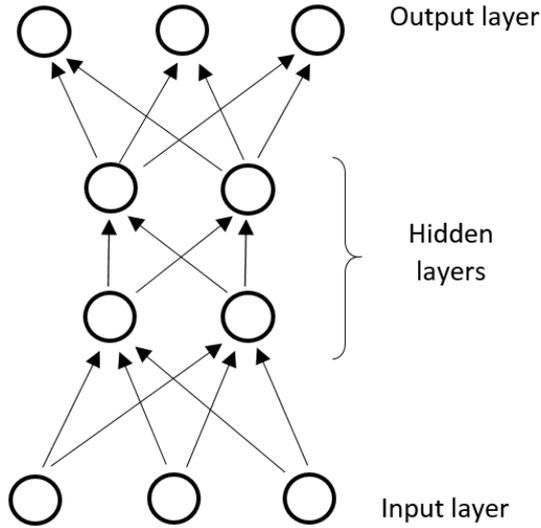


Figure 2: Visual representation of a basic MLP. Weights are represented by arrows.

2.2.1 Why an MLP?

The main reason for the choice of an MLP, is the contrast with the ESN. Since in this paper it is researched whether it is possible to decode the location of a source from the excitation pattern along a lateral line using neural networks, it is important that the chosen networks are different.

The MLP differs from the ESN mainly on two fronts. Firstly, since an MLP does not have recurrent connections, it has no short term memory. Secondly, all weights of an MLP are trained during training.

2.2.2 Specifications

The MLP used in this paper is a fully connected network. As with the ESN, the size of the input layer for the MLP is variable (32 or 64 nodes) and the output layer consists of two nodes. The network has two hidden layers, consisting of 5 nodes each.

Equation 2.1 shows the way in which the acti-

uations of the nodes are computed.

$$x_i^{m+1} = f \left(\frac{1}{N^m} \sum_{j=1}^{N^m} w_{ij}^m x_j^m \right) \quad (2.1)$$

In this equation, x_i^m denotes the activation of the i th node of the m th layer, where counting starts at the input layer. w_{ij}^m denotes the weight connecting the j th node in the m th layer to the i th node in layer $m + 1$. N^m is the number of nodes in the m th layer. f is the activation function, which is $f(x) = \tanh(x)$ when the activation of the hidden nodes is calculated. This is a typical activation function for neural networks (e.g. in Jaeger (2002)). For the output nodes, the identity function is used as activation function. The argument of f is normalized by dividing it by N^m to avoid extreme activations of the hidden and output nodes. Extreme activations in hidden nodes are likely to cause extreme activations in the output layer as well. Of course, extreme output values should be avoided.

The network was trained with the incremental learning version of the backpropagation algorithm (Rumelhart et al., 1985), using a learning rate of 0.05. All weights are initialized uniformly between -0.1 and 0.1.

2.2.3 Overfitting

The number of training epochs of the MLP was chosen so that the performance is as good as possible, while overfitting does not occur. To do this, after every epoch the performance of the network was tested on a validation set, which was constructed in the same way as the test set.

During training, the Mean Squared Error (MSE) on the training and validation set was monitored. This is the error that is minimized during training. The MSE for a network on a specific data set is given in equation 2.2.

$$\text{MSE} = \frac{1}{T} \sum_{n=1}^T \|d(n) - y(n)\|^2 \quad (2.2)$$

In this equation T is the number of samples in the data set, d is the target output and y is the network output.

2.3 Data sets

The lateral line organ consists of an array of neuromasts. Each neuromast contains hair cells that detect the movement of the water at the location of that neuromast (Flock and Wersäll, 1962). The data sets used in this research resemble the information perceived by the lateral line organ. The sets consist of simulated data. For the construction of the data sets the fluid velocities caused by a sphere moving on a 2D plane in a 3D volume through water is calculated. Figure 3 shows this setup.

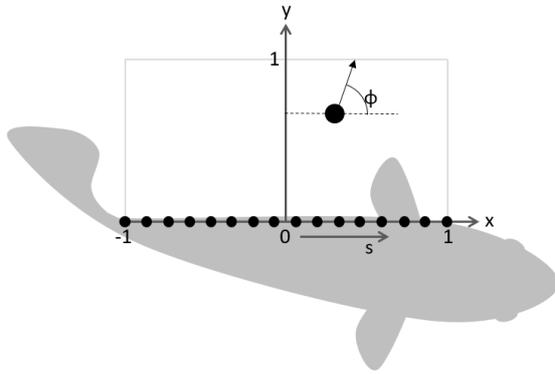


Figure 3: Model setup. The sphere moves on an x,y -plane in a direction that has angle ϕ with the lateral line. The position of a sensor on the lateral line is denoted by s .

The remainder of this section describes the way the sphere moves, the way the fluid velocities on the lateral line caused by this movement are computed and the way in which these velocities are used as input for the neural networks.

2.3.1 Movement of the sphere

The sphere starts moving at time $t = 0$ at a random location (x,y) in the Cartesian system where x is taken from a uniform distribution with range $[-1, 1]$ and y is taken from the uniform distribution with range $[0, 1]$. The sphere

is thus located in a 2×1 area to one side of the lateral line (see figure 3).

Because the state and therefore the output of an ESN is determined by the data presented in the current and previous time steps, its performance drops when there is no relation between the current and previous data point. Therefore, if the data consists of fluid velocities, caused by a sphere that too suddenly moves to a new random location every time step, it performs badly. To prevent this effect, a sphere moving with a variant direction is simulated.

The sphere moves with a constant speed of magnitude 0.05. The direction ϕ_t in radians in which the sphere moves changes according to equation 2.3.

$$\phi_{t+1} = \phi_t + A \quad (2.3)$$

Here, A , which is also in radians, is taken from the uniform distribution with range $[-1, 1]$. The change in angle per time step is limited, so changes in direction are less abrupt. This also makes the motion more realistic.

The next location at time $t + 1$ is found by moving the sphere in the direction denoted by angle ϕ_t , except when this means that the sphere moves outside of the 2×1 area boundaries. In this case, the movement direction is determined by reflecting it on the to be crossed boundary as shown in figure 4. Afterwards the sphere is moved in the resulting direction. If the sphere would still cross a boundary after the reflection, the movement direction is reflected again.

Note that the location/angle pair from which fluid velocities are computed usually contains the angle that points to the next location of the sphere. Especially when the sensors of an artificial lateral line can sample input at large frequencies, this is a realistic input combination.

The described way of movement correlates subsequent data points. This might be of help to the ESN, since this network has short term memory.

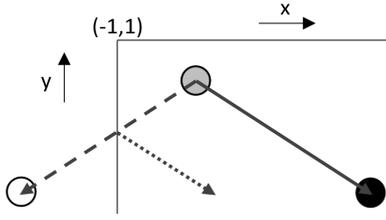


Figure 4: When the sphere moves in a direction (broken arrow) from a location (gray sphere) to another location (white sphere), but would cross a boundary, the sphere's movement direction is reflected (dotted arrow). The sphere is now moved to a location (black sphere) in the new direction (solid arrow).

2.3.2 Computing the water velocities

The simulated artificial lateral line runs from the coordinate $(-1,0)$ to $(1,0)$. The sensors are equally spaced along this line, where the first sensor is located at coordinate $(-1,0)$ and the last sensor at $(1,0)$.

In Ćurčić-Blake and van Netten (2006) it is shown that the pressure field distribution of a vibrating sphere at a given location can be calculated for inviscid low Reynolds flow. The results are in line with hydrodynamic responses. As is shown in the appendix, this pressure field distribution corresponds to the fluid velocity distribution that is shown in equation 2.4. In this equation, the sign flip suggested in the appendix is applied.

In Franosch et al. (2005) a method to find the fluid velocity distribution for a moving sphere with a constant speed is explained. This method too is only valid for inviscid low Reynolds flow.

In the appendix it is proven that the velocity field distribution described in that paper is equal to the distribution shown in equation 2.4. This equation is used to determine the fluid velocities, given the location and direction of the sphere.

$$v(s) = C(\psi_o \sin \phi - \psi_e \cos \phi) \quad (2.4)$$

v denotes the fluid velocity on position s on the

lateral line. ϕ is the angle of the source with respect to the sensor array in radians (see figure 3).

The even wavelet ψ_e and the odd wavelet ψ_o are described respectively by equations 2.5 and 2.6.

$$\psi_e(s, x, y) = \frac{1 - 2 \left(\frac{s-x}{y} \right)^2}{\left[1 + \left(\frac{s-x}{y} \right)^2 \right]^{\frac{5}{2}}} \quad (2.5)$$

$$\psi_o(s, x, y) = \frac{-3 \left(\frac{s-x}{y} \right)}{\left[1 + \left(\frac{s-x}{y} \right)^2 \right]^{\frac{5}{2}}} \quad (2.6)$$

Here (x,y) denotes the coordinate of the moving sphere.

Equations 2.5 and 2.6 show that the even and odd wavelets are dimensionless. They also show that the shape of the even and odd wavelets solely depends on the location of the sphere with respect to the sensor location. The shape of these wavelets is shown in figure 5.

In Ćurčić-Blake and van Netten (2006, pp. 1551) it is stated that the spatial variations along the x direction, as described by the even and odd wavelet functions, scale linearly with the distance of the source y . It is also stated that the maximum amplitude of the even wavelet is reached at the point of the lateral line that is closest to the source and that the odd wavelet is zero at this position. This is the location on the lateral line that is equal to the x coordinate of the sphere.

The value of C for a moving source is given in equation 2.7.

$$C = \frac{W a^3}{2y^3} \quad (2.7)$$

In this equation, W is the absolute velocity of the sphere and a is the radius of the sphere.

To ensure sensible output of the neural networks, the fluid velocities in the excitation patterns are scaled, so that the largest magnitude

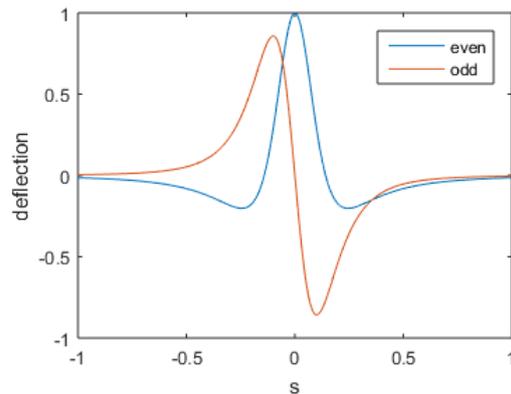


Figure 5: The shapes of the even and odd wavelets for a moving source located at coordinate (0, 0.2). The shapes are defined in equations 2.5 and 2.6 respectively.

in an excitation pattern becomes 1 (see section 2.3.3). Therefore C is discarded when computing the excitation patterns. This can be done because the (x, y) coordinate of the source is encoded in spatial variations of w along the x axis.

2.3.3 Scaling the input

Before the input is presented to the networks via the input layer, it is first scaled so that the largest magnitude in the excitation pattern becomes 1. For this, all excitation pattern values are changed according to equation 2.8.

$$\text{new } q[n] = \frac{q[n]}{\max|q|} \quad (2.8)$$

The discrete signal q is the excitation pattern and index n is the sensor number. This scaling causes the q to always be within the range $[-1, 1]$. Input patterns varying in magnitude according to C (equation 2.7) turned out to result in poor performance for both networks.

2.3.4 Splitting water velocities into two input nodes

Each data point in the data sets represents the scaled fluid velocities detected by sensors that

are equally spaced along a 1D artificial lateral line. This array of fluid velocities is used as input for the ESN and MLP. In the input layers of these networks, every calculated velocity is represented as a pair of values. This is done because the detected fluid velocities are signed values. Feeding input containing both positive and negative values to the neural networks, resulted in poor performance. This issue is resolved by observing the innervation of the hair cells in neuromasts.

In neuromasts, two types of hair cells are present. The first type only detects water flow in one direction, while the second type only detects flow in the opposite direction (Flock and Wersäll, 1962). The two sorts of information perceived by the two types might be transmitted separately to the central nervous system (Münz, 1989, pp. 290). If this is so, the fish is able to discern between positive and negative fluid velocities. In the data sets, this extra piece of information is imitated; each sensor is represented by two values. The first value only represents positive velocities. It is zero for negative velocities. For the second value, the same applies, but vice versa. Since lateral lines with 16 and 32 neuromasts are simulated, the resulting input layers consist of 32 and 64 nodes.

2.3.5 Adding noise

To test the noise robustness of the networks, different levels of noise are added to the fluid velocities computed for test sets. This way, noisy measurements of the velocities are simulated. The noise is added after scaling (section 2.3.3), but before the splitting (section 2.3.4) of fluid velocities is applied. Noise is only added to the test set.

The noise is taken from a normal distribution with a mean of zero. For different noise levels, the standard deviation of the noise is varied. The noise level is defined in this paper in terms of the Signal Noise Ratio (SNR). The definition of the SNR is given in equation 2.9.

$$\text{SNR} = \frac{A_e}{\sigma_n} \quad (2.9)$$

In this equation A_e is the maximal magnitude of deviation from zero of an input excitation pattern e . σ_n is the variance of the distribution out of which the noise values are taken.

The quantities of noise in the excitation patterns might not always be known. To simulate unknown quantities of noise, the SNR is pretended to be unknown during the testing of the networks. However, directly after adding noise, information about the quantity of the noise is present in the input pattern amplitude. To remove this information, the input pattern is scaled again in the way described in section 2.3.3.

2.3.6 Paths of sphere movement in the data set

The ESN and MLP are trained using a training set and a test set. With the MLP, a third set of data is also used to avoid overfitting during training (see section 2.2.3).

One path for the movement of the sphere was used for training sets, one for test sets and one for validation sets. This choice was made because it is assumed that the paths contain enough samples to represent the population of possible source locations and directions.

However, in the paths, some source locations and directions may be better represented than others. Because of this, the check for overfitting in the training stage of the MLPs is done.

The sets used contain 5000 samples each. A large number of samples was chosen to make sure that the possible source locations were well represented.

3 Results

3.1 Network output used for comparing

To see the differences in performance between the different kinds of networks and different ar-

ray sizes, 150 ESNs and 30 MLPs for both input excitation pattern sizes were trained. A larger amount of ESNs was trained, because they can be trained faster and because the variance of the error on the output of the ESNs was greater.

The random initialization of the weights of the networks and the different validation sets resulted in different networks per combination of network type and input size. Using multiple networks per combination of input size and network type, maps this variety to the output error.

To denote the set of networks that gets an input resulting from a specific amount of sensors, a network type and a number of sensors in subscript is used. E.g. ESN₁₆ denotes the set of ESNs to which excitation patterns with 16 sensors (resulting in input layers with 32 nodes) are presented.

3.2 General network output

Figure 6 gives a typical indication of the performance of the networks when no noise is added to the input.

3.3 Overall performance

For comparing the difference in performance between different kinds of networks, an error measurement is needed that shows the overall error of a network. To get such an error measurement, the Mean output error in Euclidean Distance (MED) is computed for every network in the sets that represent combinations of a network type and input size.

The boxplots in figure 7 show the MED distribution per network type. Figures 8 and 9 show the error of the network types on respectively the x and y coordinates. The boxplots are constructed with network output given input with no added noise.

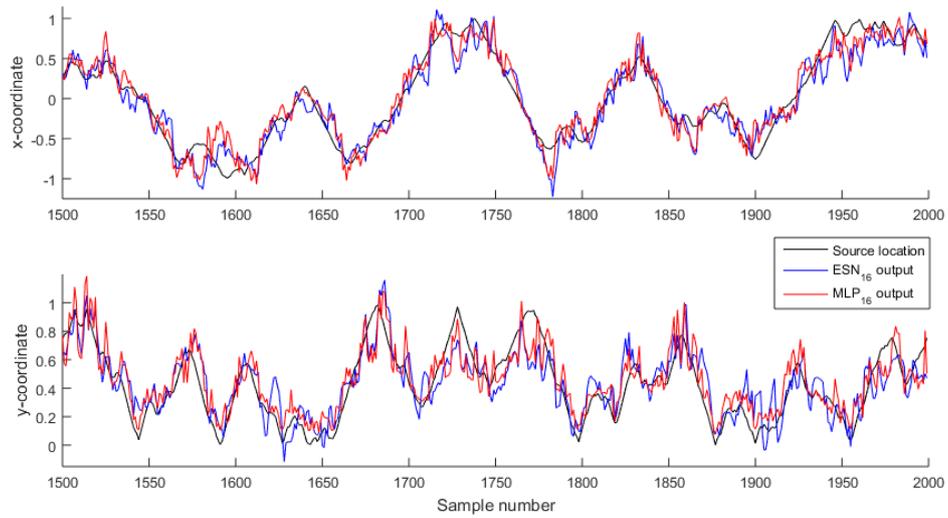


Figure 6: Actual source location and output location on a part of the test set of individual networks of the ESN₁₆ and MLP₁₆ sets.

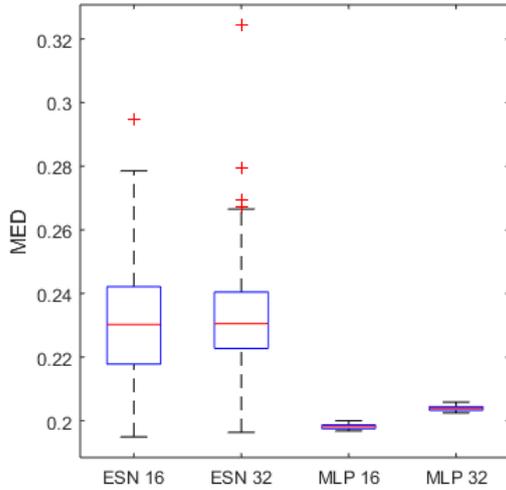


Figure 7: Boxplots of the MED distribution per network type

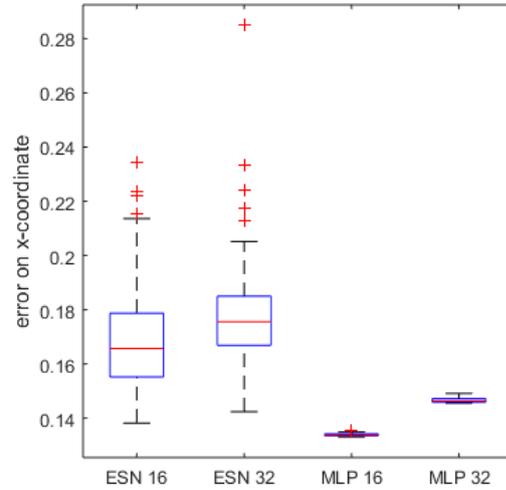


Figure 8: Boxplots of the error on the x coordinate per network type

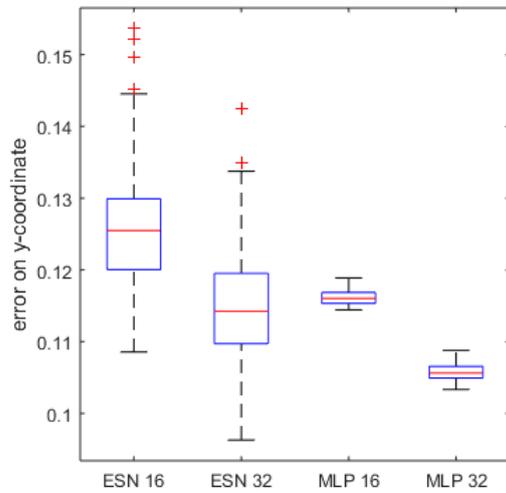


Figure 9: Boxplots of the error on the y coordinate per network type

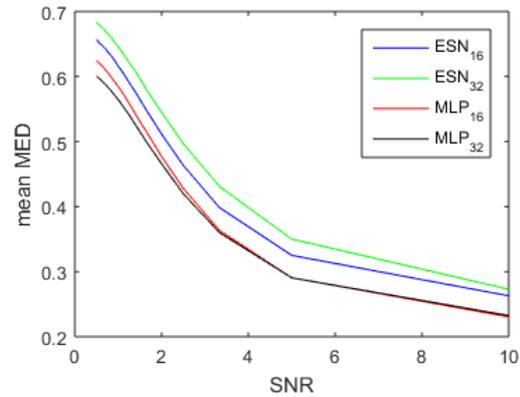


Figure 10: This graph shows the noise robustness of the performance in MED of the different kinds of networks.

3.4 Noise robustness

The MED of each type of network was found for different SNRs. The SNRs were sampled in the interval $[0.5, 10]$. The results are plotted in figure 10. Figures 11 and 12 show the noise robustness of determining respectively the x and y coordinates of the sphere.

3.5 MLP overfitting

When overfitting occurs, a model starts looking at specific situations instead of general ones. This causes the model to perform better on the training set, but worse in the general case. To make sure this does not happen, for all trained MLPs, the MSEs on the training set were plotted with the MSEs on the testing set for every epoch during training. An example of such a plot can be seen in figure 13. For no MLP, the MSE of the test set increased continuously, or to a stable value at any point in training. Therefore, overfitting does not occur.

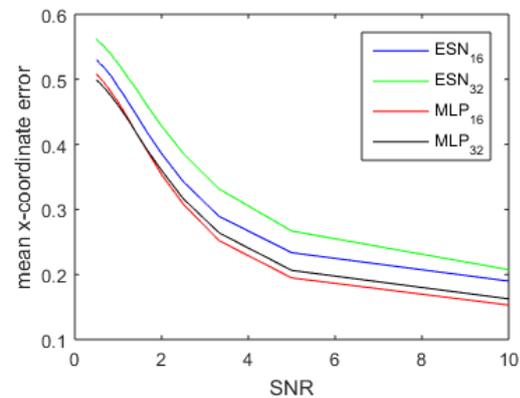


Figure 11: The noise robustness on determining the x coordinate of the sphere.

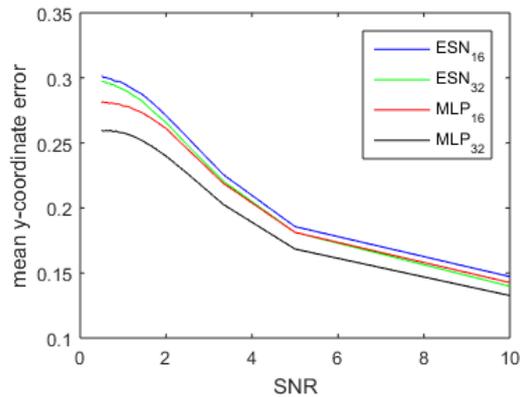


Figure 12: The noise robustness of the of determining the y coordinate of the sphere.

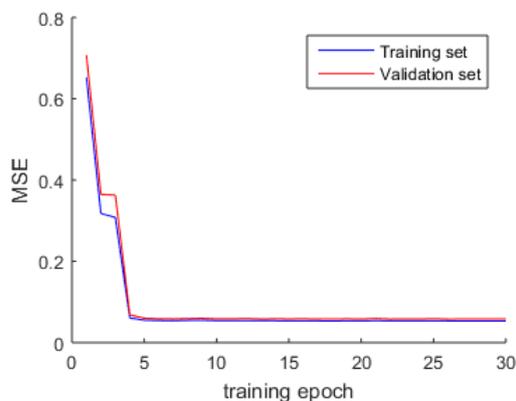


Figure 13: An example of the MSE on the validation set and on the training set.

4 Conclusion

4.1 Overall performance

From the results in figure 6 it can be concluded that neural networks are able to extract information about the location of a moving source. The network with the highest accuracy produced output that had an MED of less than 0.2 in a field with dimensions 2×1 .

On average, the MLP performed significantly better than the ESN even though it did not use information about past input. The MLPs with 16 sensors performed significantly better than

the MLPs with 32 sensors.

However, when a network type is represented by its best performing individual network, the ESN with 16 sensors performed best.

4.2 Variance in ESN output error

Figures 7, 8 and 9 all show clearly that the variance of the output error is much larger for the ESN than for the MLP. During training, all weights of the MLP are trained, while only the weights to the output layer are trained with the ESN. Therefore a lot of randomly initialized weights are still random when the ESN is exploited. If some weight configurations are better than others, this causes variance in the ESN performance.

4.3 Difference in performance on x and y coordinates with respect to the number of sensors

Figure 8 and 9 show that there is a clear difference between the output error on the x and y coordinates.

There is only little overlap between the ranges of error of the two coordinates. The error on the y coordinate is clearly lower on average. This was expected, since in the data sets, the range of possible y coordinates was half the range of the x coordinates.

On determining the y coordinate, networks with 32 sensors performed better for both network types. The best instance was from ESN₃₂. The bottom right plot in figure 14 shows that the difference in performance between 16 and 32 sensors on the y coordinate originates from the samples where the sphere is close to the sensor array.

Networks with 16 sensors performed better on determining the x coordinate than networks with 32 sensors. The top left and top right plots in figure 14 show that the difference in error on respectively the x coordinate for 16 and

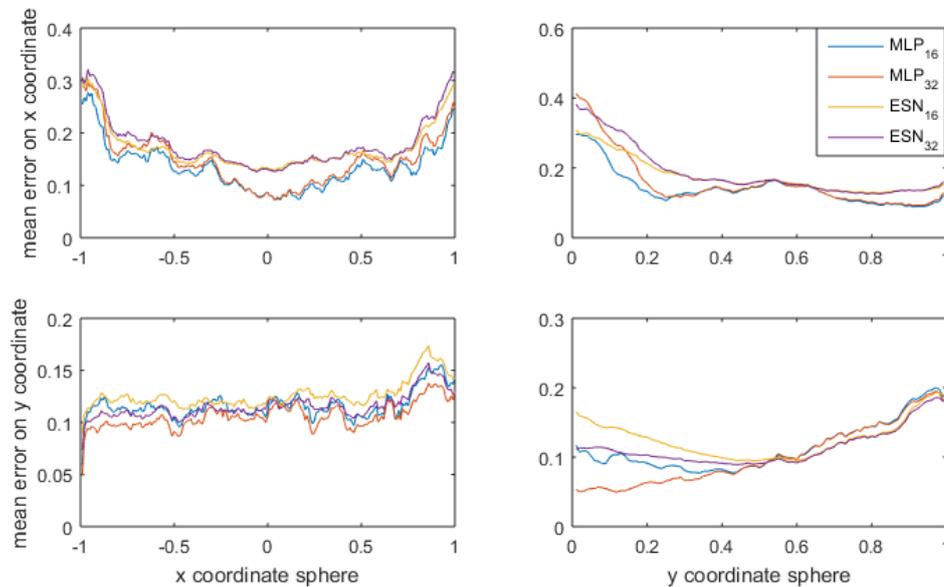


Figure 14: Separate error of the networks on x and y coordinates versus the coordinates of the sphere

32 sensors is dependent on the x and the y coordinate of the sphere. The difference occurs when y is close to 0 and x is close to -1 or 1. This is clearly visible in figure 15.

4.4 Network output location

In figure 16 it is apparent that the networks, especially the MLP, never produce output in the corners of the area in which the sphere moves, even though the corners are well represented by the training and test data.

Figure 16 also shows that MLPs with 16 sensors give no output at very low values of y . The same does not apply for MLPs with 32 sensors. Therefore, this result might be explained by the larger distance between the sensors. In Ćurčić-Blake and van Netten (2006, pp. 11) it is shown that a source should not be closer than approximately twice the inter-sensor distance to be detected. For a lateral line with length 2, this minimum distance is $\frac{2}{16} * 2 = 0.25$ for an array of 16 sensors and $\frac{2}{32} * 2 = 0.125$ for an array of 32 sensors.

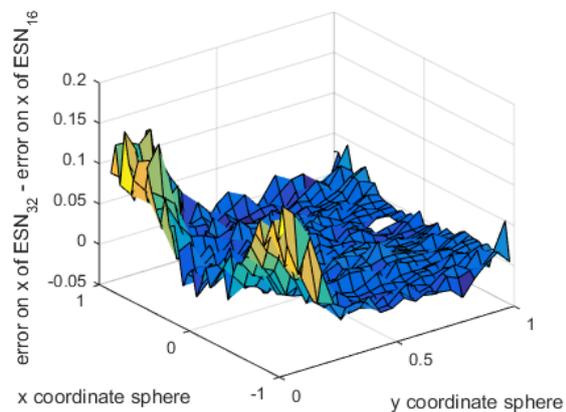


Figure 15: Difference between the error on the x coordinate for ESN₁₆ and ESN₃₂ given the location of the sphere in the 2×1 area. For the difference between the error on the x coordinate for MLP₁₆ and MLP₃₂ a similar pattern is found.

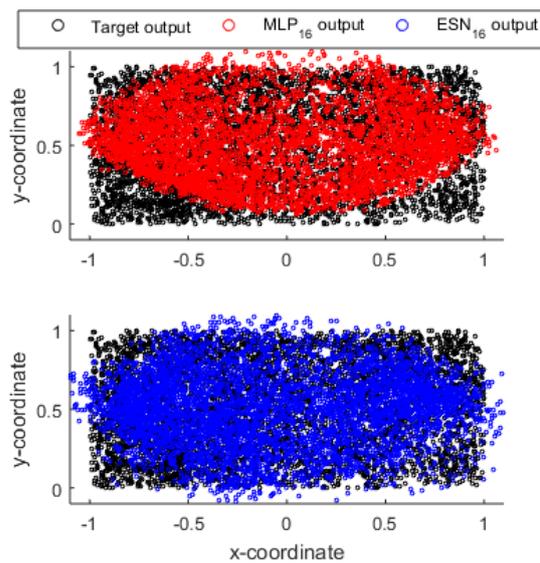


Figure 16: The coordinates of instances of the ESN₁₆ and MLP₁₆ output versus the coordinates of the target output.

4.5 Decreased performance in the corners of the field

Figures 14 and 16 indicate an overall decrease of performance in the corners of the field in which the sphere moves.

When the y coordinate of the sphere is close to 1, the wavelets as shown in figure 5, and thus wavelet w , described in 2.4, become wide enough so that their shapes can only be partially detected by the sensors along the lateral line. The shape of w is also only partially detected when the source moves close to either side of the field, because the center of w is equal to the x coordinate of the sphere. An input excitation pattern that is constructed with only a part of the shape w could cause a decrease in performance.

Furthermore, when the sphere is close to the sensor array, the shape of w becomes narrow. A narrow shape is described in less detail in the input excitation pattern, since the shape is sampled by less sensors. This could cause the decrease in performance when the sphere moves close to the array.

Combinations of the effects described above might be partially responsible for the decrease in performance when the sphere moves near the corners of the field.

4.6 Noise robustness

All networks show similar noise robustness. The error grows slowly to when the SNR is about 5. For lower SNRs, it starts growing faster.

The scaling of the input (section 2.3.3) might have a negative effect on the noise robustness. If the largest magnitude of an input pattern becomes greater when noise is added, the original signal is flattened when the pattern is rescaled adding the noise. This adds another type of deformation to the signal.

The SNR is linearly proportional to the amplitude of the excitation patterns presented to the network. This amplitude is linearly proportional to C (equations 2.7 and 2.4). Therefore the SNR is, like C , linearly proportional to the velocity W of the source and inversely proportional to the cubed distance from the source to the sensor array (y^3). Far away objects are thus difficult to detect with this method, even when only little noise is present.

There was no large difference found in the effect of noise for the output error on the x and y coordinate. The larger error on the x coordinate than on the y coordinate for low SNRs can be attributed to the dimensions of the field in which the sphere moved.

5 Discussion

5.1 Further research to single source localization

The current method can certainly be improved. It is improbable that the performance of the networks in this research reflects the optimal performance for localizing a source using lateral line excitation patterns. The reason for this is

that it is uncertain whether the chosen parameter settings for the networks or the number of input sensors were close to optimal. Extending the input with indicators might also improve the network performance. These indicators can hold information about e.g. the width of the wavelet or the location of minima and maxima in the excitation pattern. There might even be other neural networks or completely different methods that are better suited for the task at hand.

Furthermore, differences in performance between different network types for specific locations of the source have been found that have not yet been explained. Examples of this are the difference in error between the ESN and MLP on the x coordinate for small and large magnitudes of the x coordinate of the source and on the y coordinate for small and large values of the y coordinate of the source (see figure 14).

When the source is close to the lateral line, performance of networks with 32 sensors is lower than that of networks with 16 sensors. The networks are thus hindered by the extra information that is available when the extra sensors are used. To utilize this extra information for source localization, transformations of the input signal or the use of different types of networks or methods might be needed.

5.2 More complex forms of underwater source detection

In this research, the network output only gives the 2D location of a source, because it only uses water velocities in the direction parallel to the lateral line. If a gradient in the orthogonal direction is also used as input, the output can be changed to the location of a moving source in a 3D volume. This orthogonality of neuromast arrays can also be found on the heads and sometimes along the sides of fish (Coombs et al., 1988, pp. 568-576). On the heads, canals with neuromasts and arrays of superficial neuromasts with different angles to each other are present. This can allow for 3D source local-

ization around the head of the fish. Also, the lateral line along the side of some fish has a zig zag shape, so that parts of the lateral line are almost or entirely orthogonal to other parts of the line.

Secondly, this method cannot easily be used for detecting multiple sources, although it can be altered in at least two ways that might make multiple source localization possible. Firstly, a network for every plausible amount of moving sources could be trained. A problem of this approach is that an accurate criterion for which network output should be used is needed. Secondly, instead of producing the coordinates of the source, the networks could be trained to output a 2D grid in which only nodes that represent locations where a moving source is present have large activation values.

The outcomes of this research are promising, but there is still much research required before an artificial lateral line can be produced that can image underwater environments in high resolution.

References

- Sheryl Coombs, John Janssen, and Jacqueline F Webb. Diversity of lateral line systems: evolutionary and functional considerations. In *Sensory biology of aquatic animals*, pages 553–593. Springer, 1988.
- Branislava Ćurčić-Blake and Sietse M van Netten. Source location encoding in the fish lateral line canal. *The Journal of experimental biology*, 209(8):1548–1559, 2006.
- Åke Flock and Jan Wersäll. A study of the orientation of the sensory hairs of the receptor cells in the lateral line organ of fish, with special reference to the function of the receptors. *The Journal of Cell Biology*, 15(1): 19–27, 1962.
- Jan-Moritz P Franosch, Andreas B Sichert, Maria D Suttner, and J Leo van Hemmen. Estimating position and velocity of a submerged moving object by the clawed frog

Appendix

xenopus and by fisha cybernetic approach. *Biological cybernetics*, 93(4):231–238, 2005.

Denise Hoekstra and John Janssen. Non-visual feeding behavior of the mottled sculpin, *cottus bairdi*, in lake michigan. *Environmental biology of fishes*, 12(2):111–117, 1985.

Herbert Jaeger. *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach*. GMD-Forschungszentrum Informationstechnik, 2002.

LAKHsMI. Sensors for large scale hydrodynamic imaging of ocean floor, 2015. URL http://cordis.europa.eu/project/rcn/193338_en.html.

Heinrich Münz. Functional organization of the lateral line periphery. In *The mechanosensory lateral line*, pages 285–297. Springer, 1989.

Brian L Partridge and Tony J Pitcher. The sensory basis of fish schools: relative roles of lateral line and vision. *Journal of Comparative Physiology*, 135(4):315–325, 1980.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.

Equivalence between wavelets as described by Curcic and van Netten (2006), and Franosh et al. (2005).

SvN, 13th November 2015.

From Franosh et al. 2005, or originally Lamb (1932), we take the velocity potential at location \mathbf{r} of a moving sphere traveling with speed \mathbf{W} , and considered to be instantaneously located in the origin ($\mathbf{r}=0$): $\phi = \frac{a^3}{2r^3}(\mathbf{W}\mathbf{r})$. Taking the negative gradient in just the x-direction yields the fluid velocity along the x-axis generated by the moving sphere when at location $\mathbf{r}=(x,y,0)$. Here the problem is thus reduced from 3D to 2D by taking $w_z=z=0$ (so $\mathbf{W} = W_x\hat{\mathbf{x}} + W_y\hat{\mathbf{y}}$; $\mathbf{r} = x\hat{\mathbf{x}} + y\hat{\mathbf{y}}$). Then evaluating in Cartesian coordinates we get the component of fluid velocity along the x direction, as this we'll take to be the direction along which the array is situated (without loss of generality):

$$v_x = -\nabla\phi \cdot \hat{\mathbf{x}} = -\frac{d\phi}{dx} = -\frac{d}{dx}\left(\frac{a^3}{2r^3}(\mathbf{W}\mathbf{r})\right) = -\frac{d}{dx}\left(\frac{a^3}{2} \cdot \frac{W_x x + W_y y}{(x^2 + y^2)^{\frac{3}{2}}}\right) =$$

$$-\frac{a^3}{2} \cdot \left[-\frac{3W_x x + W_y y}{2(x^2 + y^2)^{\frac{5}{2}}} \cdot 2x + \frac{W_x}{(x^2 + y^2)^{\frac{3}{2}}} \right] = \frac{a^3}{2} \left[\frac{3(W_x x + W_y y)x - W_x \cdot (x^2 + y^2)}{(x^2 + y^2)^{\frac{5}{2}}} \right].$$

Now, introducing the position variable s along the array, we put the sphere's horizontal position along this array at b (so $x=s-b$) and at a vertical distance d (so $y=-d$) from the array (i.e. similarly defined as in Curcic and van Netten (2006)). Then we have for the horizontal fluid velocity profile, $v(s,b,d)$, along the array at position s , and on the moment that the sphere is at location (b,d) :

$$v(s, b, d) = \frac{a^3}{2} \left[\frac{3(W_x(s-b) - W_y d)(s-d) - W_x \cdot ((s-b)^2 + d^2)}{([s-b]^2 + d^2)^{\frac{5}{2}}} \right].$$

This may be rewritten in dimensionless form in between the square brackets as:

$$v(s, b, d) = \frac{a^3}{2} \left[\frac{3d^2(W_x(\frac{s-b}{d}) - W_y)(\frac{s-d}{d}) - W_x d^2 \cdot \left(\left(\frac{s-b}{d}\right)^2 + 1\right)}{d^5 \left(\left[\frac{s-b}{d}\right]^2 + 1\right)^{\frac{5}{2}}} \right] = \frac{a^3}{2d^3} \left[\frac{3\left(W_x\left(\frac{s-b}{d}\right)^2 - W_y\left(\frac{s-d}{d}\right)\right) - W_x \cdot \left(\left(\frac{s-b}{d}\right)^2 + 1\right)}{\left(\left[\frac{s-b}{d}\right]^2 + 1\right)^{\frac{5}{2}}} \right].$$

Separating the terms with the sphere's two velocity components and writing for these, $W_x = W \cos(\varphi)$ and $W_y = W \sin(\varphi)$, where W makes an angle φ with the array and where $W = |\mathbf{W}|$ we arrive at:

$$v(s, b, d) = \frac{W a^3}{2d^3} \left[\frac{\cos(\varphi)\left(2\left(\frac{s-b}{d}\right) - 1\right)}{\left(\left[\frac{s-b}{d}\right]^2 + 1\right)^{\frac{5}{2}}} - \frac{\sin(\varphi)3\left(\frac{s-d}{d}\right)}{\left(\left[\frac{s-b}{d}\right]^2 + 1\right)^{\frac{5}{2}}} \right], \text{ so}$$

$$v(s, b, d) = \frac{W a^3}{2d^3} [-\Psi_e \cos(\varphi) + \Psi_o \sin(\varphi)] \quad (1).$$

Here we have used the same definitions of the even and odd dipole wavelets Ψ_e and Ψ_o as used in Curcic and van Netten (2006); we observe the same linear scaling properties here. Note that here the velocity profile of the fluid flow component along the array is calculated in response to a sphere with velocity \mathbf{W} .

In Curcic and van Netten (2006), the pressure gradient along an array was derived as being produced by a vibrating sphere at location (b,d) moving with vibratory displacement $x(t) = X_0 \sin(\omega t)$, subtending an angle φ with the array, and resulting in a vibratory pressure gradient $\frac{dp(t)}{ds} = \frac{dP}{ds} \sin(\omega t)$. There it was found that then:

$$\frac{dP}{ds} = \frac{\rho\omega^2 a^3}{2d^3} X_0 [\Psi_e \cos(\varphi) + \Psi_o \sin(\varphi)] \quad (2).$$

That approach, to determine the pressure gradient as the signal relevant for lateral line processing, was taken as it is known from lateral line experiments that canal cupulae are displaced in proportion to local water acceleration $a(t)$, which is proportional to the pressure gradient under free fluid conditions:

$a(t) = \frac{-1}{\rho} \frac{dp(t)}{ds}$. so that fluid velocity as a function of time equals:

$$v(s, t) = \int a(t) dt = \int \frac{-1}{\rho} \frac{dp(t)}{ds} dt = \frac{-1}{\rho} \int \frac{dP}{ds} \sin(\omega t) dt = \frac{1}{\rho\omega} \frac{dP}{ds} \cos(\omega t) \quad (3).$$

The motion of the vibrating sphere in Curcic and van Netten (2006) is described as: $x(t) = X_0 \sin(\omega t)$, so therefore its (vibratory) velocity is $W(t) = X_0 \omega \cos(\omega t)$. To compare both situations we now consider the vibrating sphere situation in terms of its velocity $W(t)$. Rewriting (3) with the use of (2) yields (with introducing the position (b,d) of the sphere as extra parameters):

$v(s, b, d, t) = \frac{1}{\rho\omega} \frac{dP}{ds} \cos(\omega t) = \frac{1}{\rho\omega} \frac{\rho\omega^2 a^3}{2d^3} X_0 \cos(\omega t) [\Psi_e \cos(\varphi) + \Psi_o \sin(\varphi)]$. In terms of velocity W this gives (while dropping the time parameter t again):

$$v(s, b, d) = \frac{W a^3}{2d^3} [\Psi_e \cos(\varphi) + \Psi_o \sin(\varphi)] \quad (4).$$

This is equivalent to Eq. (1) There only seems to be a minus sign discrepancy in (4) as compared to (1) in relation to the even wavelet term (I have to figure this out, but suspect now there is a sign error in the Curcic and van Netten 2006 paper).

Conclusion:

The most important conclusion from the above is that there is **no difference** between both the even and odd wavelet shapes derived from a pressure field distribution of a **vibrating sphere ($x(t) = X_0 \sin(\omega t)$)** and, on the other hand, starting from a general velocity potential function, of a **sphere moving with constant velocity (W)**. Both situations, though, are only valid for inviscid (i.e. irrotational) low Reynolds (Stokes) flow. This regime looks most favorable for applications using an array for hydrodynamic imaging. For the case that a vibratory sphere is considered in a viscous fluid, but flow is measured outside its boundary layer, we can still make use of the above results but have to take a time (i.e. phase) shift into account. Also, we have to further look into situations with higher (flow) velocities, i.e. for high Reynold numbers, where vorticity also starts to emerge in the sphere's wake.