The influence of fluid model complexity on optimal sensor configuration for object localization in water

Bachelor's Project Thesis

Niek Naber, n.naber@student.rug.nl,
Supervisors: dr. S.M. van Netten & B.J. Wolf, MSc

Abstract: The lateral line is an organ which fish use to sense their surroundings. Lateral lines consist of several neuromasts which serve as sensors of fluid flows. These fluid flows are used to make estimates about objects in the fish's surroundings. Artificial lateral lines can also be used to estimate objects positions and orientations in water. Different simulation methods exist to simulate water, where some do include viscosity, and others do not. This study researches differences between these simulation methods, the way extreme learning machines need to be trained on fluid velocities of the simulations to predict an objects location, the prediction errors of those extreme learning machines and the optimal sensor positions assigned by a genetic algorithm. Because of the noisy effects of viscosity, extreme learning machines that use viscous simulations need more training samples to reach their optimal performance. Their optimal performance is worse than that of extreme learning machines that use non-viscous simulations, especially when grid-like sensor set ups are used. The extreme learning machines that use viscous simulations seem to benefit more when fluid velocities are sensed at a lower height below objects, and when sensors are placed more to the outside of a sensor plane, as compared to a grid like setup. When an extreme learning machine is trained on either one of the simulations, there seem to be major differences in prediction errors when we vary the positions that are used to measure the fluid velocities. When a genetic algorithm is used to find the sensor configuration with which the extreme learning machine has the lowest prediction error, sensor positions seem to be pushed to the sides of the sensor plane. This study had limited computational and storage capacity, therefore findings can be improved by further research which takes higher resolution simulations, and bigger sample sizes.

1 Introduction

Fish can use an organ called the lateral line to detect fluid flows around them (Dijkgraaf, 1963). Lateral lines consist of multiple neuromasts. A neuromast is an organ with which fluid flow is measured (Horst Bleckmann and Hanken, 2004). These neuromasts therefore can be considered fluid flow sensors. With a lateral line organ fish can estimate positions of moving objects that are near the fish (Dijkgraaf, 1963).

It is possible to use these techniques that fish are using in artificial applications, as is shown by B. Curcic-Blake and S.M. van Netten (2006). In an artificial application, an array of one- or two-dimensional sensors can pick up fluid flows of the water. Several artificial networks have been trained on the data of such arrays to detect the source location and direction based upon the output of a sensor array (Wolf and van Netten, 2018). The application of an artificial lateral line (ALL) is useful for example for source detection by underwater vehicles, when other sensor that rely on vision or sonar can not be used (Vollmayr, Sosnowski, Urban, Hirche, and van Hemmen, 2014).

With a slight alteration, an ALL might also be useful for instance in a harbor, to detect at which positions the ships are located. For this we might want to space out the sensors over a grid instead of

a single line to get the best results.

Steenkist (2019) has done his bachelor project on finding the optimal sensor configuration of an extreme learning machine in a potential flow simulation, so that the sensors have the least dark corners when detecting a moving source in water. To determine the optimal sensor configuration, he used a genetic algorithm, K-means, Kohonen SOM and basic sensor setup (Figure 3.1). He did not take viscosity of the water into account in his experiments because the simulation method potential flow does not include those factors.

1.1 Goals of this study

There exist multiple simulation methods to simulate fluid flows in water. A potential flow simulation is one of the simpler simulation methods, which does not take the viscosity of water into account.

The aim of this study is to determine the following:

- Whether viscosity has an effect on the prediction error of an extreme learning machine that is trained on fluid velocities.
- The optimal sensor positions for training an extreme learning machine on fluid velocities at those positions, to predict the locations of objects.

Several variables are to be compared to each other for two simulation methods, where one does include viscosity, and the other does not.

In this paper we will firstly discuss some theoretical background to techniques used in this study. Then some general methods are explained. After the methods, a parameter study will be performed in which multiple parameters are tested to find settings in a way such that an extreme learning machine will have the lowest mean Euclidean distance prediction error. Then a genetic algorithm will be discussed, that tries to find the sensor configuration that result in the lowest mean Euclidean prediction error of an extreme learning machine. This genetic algorithm uses the settings found in previous sections. Lastly a discussion and conclusion will follow.

2 Background

Several methods and techniques will be used in this study to generate fluid flow velocities in simulations, and predict object locations in those simulations. Potential flow is the simulation method that will be used that excludes viscosity. Manta Flow is the simulation method that will be used that includes viscosity.
After the simulations, extreme learning machines are used to predict the source location of objects in the simulations. Several parameters of the extreme learning machine will be adjusted to find optimal settings to predict object locations with fluid velocities from the two simulations.
Lastly, a genetic algorithm will try to find the most optimal sensor positions in a grid to predict the objects location, based on the prediction error expressed in a mean Euclidean distance.

2.1 Potential Flow

To simulate water flows in potential flow we need to know the source position ($\vec{q}$), sensor position ($\vec{s}$) and source speed ($\vec{w}$). The potential flow equation that is used in this study is equation 2.1. This equation is derived from the potential flow equations by Primoz and van Netten (2018). Here $\vec{r}$ is the distance from the sensor to the source, so $\vec{r} = \vec{q} - \vec{s}$. Paramater a is the source radius. $\vec{v}$ is the vector of the fluid flow at sensor position $\vec{s}$.

$$\vec{v} = \frac{-a^3}{2|r|^3}(\vec{w} - \frac{3(\vec{r}.\vec{w})\vec{r}}{|r|^2}) \tag{2.1}$$

2.2 Manta Flow

Manta Flow is a simulation framework that is written in the programming language c++. This framework allows for a more complex simulation in which viscosity is taken into account (Thuerey and Pfaff, 2018). As a result of viscosity, turbulent flows will be present in the velocity fields. Manta Flow does not include the metric system in its measurements. All measurements are done in grid points which can be used as reference to meters. The units that we use in Manta Flow are (time-)steps, grid points and frames. A frame is a plane of grid points at a certain height.

Unlike potential flow, Manta Flow does not give an instantaneous fluid flow field for a moving object. An object needs to be instantiated in a field or space, and then being moved. Once the object is being moved, fluids start to flow.

2.3 Extreme Learning Machine

In 2004, Guang-Bin Huang firstly proposed an extreme learning machine(ELM) as an alternative to single-layer feed forward neural networks(SLFN's). An ELM randomly chooses the input weights and analytically determines the output weights of SLFN's(Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew, 2004). The ELM was invented to improve training speed of a neural network because often a traditional SLFN is rather slow. Boulogne, Wiering, Wolf and van Netten (2017) showed that the performance of an ELM in localizing a moving object is at least as good as that of a multilayer perceptron, where it shows a great speed-up in training time.

2.4 Genetic Algorithm

A genetic algorithm is an algorithm designed for optimization, and is inspired by natural selection (Holland, 1962; Mitsuo Gen, 1997). A genetic algorithm starts with an initial population. The individuals in the population need to have variables or properties that can function as genes. Every individual in the population is tested against the fitness function of the genetic algorithm, and assigned a score. The top group of individuals with the highest scores of the population are picked to survive onto the next generation. The other individuals die and are neglected in the next generations.
The surviving individuals can now make new individuals by crossing over their genes. On top of the gene cross over it is also possible to include mutation of genes or other heuristics. A new population emerges and the process starts again at the selection phase where each individual is tested against the fitness function. This process repeats for a certain amount of steps, or until a certain condition is reached.

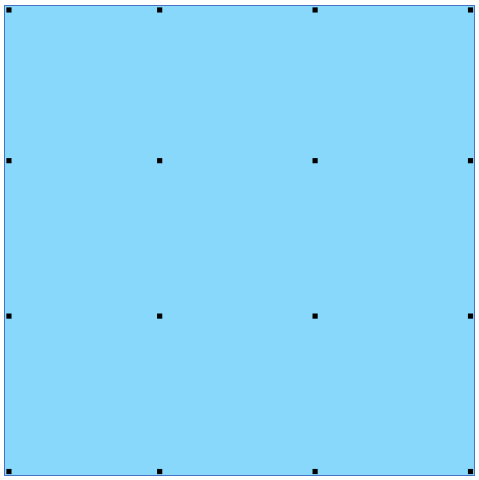

Figure 3.1: The sensor positions in the sensor plane. All black dots are sensors. Name of this setup: basic sensor setup.

3 Methods

As said before, multiple parameters are tested and compared for the two simulation methods. These parameters are:

- The number of neurons which an ELM uses in its hidden layer.
- The number of samples which are used as a training set for the ELM.
- The ideal height of the sensor plane respectively to that of the source plane for predicting locations with the use of an ELM.
- The ideal sensor positions which feed fluid velocities to an ELM to predict the object locations.

The performance of an ELM is measured by the localisation error measured in mean Euclidean distance in centimeters (MED prediction error). The input of an ELM consists of 32 data points. These points are the fluid velocities measured in x and y direction of the 16 sensors. Before fluid velocities are fed to an ELM, they are normalized. For normalization, all fluid velocities are divided by the highest encountered absolute value of all fluid velocities present in the whole set of simulations that is fed to the ELM. The values that are fed to the ELM are thus between -1 and 1. The output of the

ELM's are two values that predict the x and y coordinate of the object.

3.1 Data Generation

The data generation follows the methods of Steenkist (2019) as close as possible. We use a horizontal 2 by 2 meter source plane in which round objects with a radius of 0.06 meters can be moving with a speed of 0.13 meters per second. The x and y position of each object are randomly uniformly chosen between -1 and 1. The angle in which the objects are moving is randomly uniformly chosen between 0 and 2π. A Gaussian noise of $1 * 10^{-6}$ is added to both the x and y component of the fluid velocities. In the case of the Manta Flow simulations we need to translate the 2 by 2 source plane to a source plane with a resolution. We use a resolution of 50 by 50 grid points for this. The total simulation uses frames of 100 by 100 grid points. In this way fluid velocities can be measured outside of the source plane area, and sensors can be placed outside the source plane area. In Manta Flow simulations, the object is moving in the first time step. In the time step after that, sensor data is read from the fluid velocity field. In this way the sensor data is most similar to that of potential flow. One grid point in Manta Flow simulations used in this paper, is set to be equal to 4 centimeter. The configuration of these settings is based on the bachelor thesis of Wester (2019).

3.2 Basic Sensor Setup

To compare the multiple parameters of the ELM's that are trained on fluid velocities from different simulations, sensors need to be placed at a certain position. In all these experiments 16 sensors will generate data that is fed to an ELM. A basic sensor setup is used where sensors are evenly spread out in the sensory plane, which is located below the source plane. This sensor setup can be seen in figure 3.1. To be able to refer easily to this sensor setup we call this the *basic sensor setup*.

4 Parameters

In the upcoming sections, the two terms *potential flow data* and *Manta Flow data* will be used. With these terms we mean the fluid velocities found in the fluid flow fields in the simulation (potential flow or Manta Flow), at the specific positions indicated by the sensor positions that are used in that particular experiment.

4.1 Number of Neurons

4.1.1 Methods

To test what the optimal number of neurons in a hidden layer would be for an ELM that uses either potential flow or Manta Flow data, an ELM is trained multiple times. The height of the sensor plane relatively to the source plane is -0.25 centimeters in the potential flow simulation, and -6 frames in the Manta Flow simulation. These 6 frames correspond to 24 centimeters since each frame or grid point in our simulation can be compared to 4 centimeters: hence data cannot be read at precisely 25 centimeters because that distance would be in between frames. The plane heights are set to the value that Steenkist (2019) uses in his study. In the both simulations we used 24000 training samples and 3000 test samples, also replicating the study of Steenkist (2019).

4.1.2 Results

Figure 4.1 shows the MED prediction error of an ELM trained on potential flow data versus the number of neurons. We see a hyperbolic shape in this graph with the asymptote at about 10 centimeters for the test set. The test on the training set first also shows a hyperbolic shape but does not have a clear asymptote, it keeps linearly decreasing after the initial hyperbolic shape.

In figure 4.2 the MED prediction error ELM's trained on Manta Flow data are shown versus the number of neurons. This figure also shows a hyperbolic shape. The asymptote to which the MED prediction error converges is 30 centimeters. The MED prediction error of the training set keeps decreasing if we increase the number of neurons that we add to the hidden layer.

The ELM's seem to have reached the minimal MED prediction error when 1500 neurons are used in the hidden layer when training on potential flow data, and 1000 neurons when training on Manta Flow data.

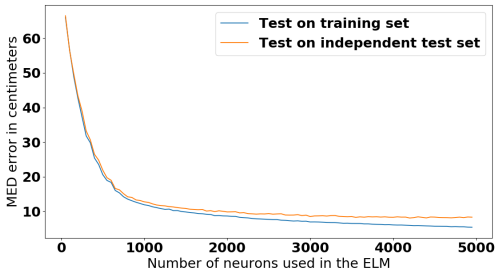

Figure 4.1: The MED prediction error versus the number of neurons used in an ELM, while trained on potential flow data. The sensor plane is positioned 25 centimeters below the source plane. For the generation of this graph an ELM's are trained 100 times, with the number of neurons in the hidden layer going from 50 to 5000 with an interval of 50 neurons.

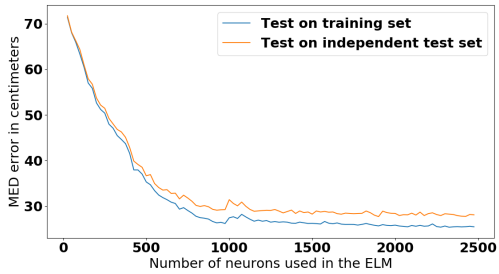

Figure 4.2: The MED prediction error versus the number of neurons used in an ELM, while trained on Manta Flow data. The sensor plane is positioned 6 frames below the source plane. For the generation of this graph an ELM's are trained 100 times, with the number of neurons in the hidden layer going from 25 to 2500 with an interval of 25 neurons.

4.1.3 Discussion

To find the optimal number of neurons to use in the hidden layer of an ELM, two aspects must be taken into account. The MED prediction error of the training set should be as low as possible, but the difference in MED prediction error between the test and training set must also be kept as low as possible. This is because the bigger this difference is, the more over-fitting is present in the ELM. The optimal number of neurons used in the hidden layer of an ELM is therefore considered to be 800 and 1000 for respectively potential flow and Manta Flow simulations. These numbers are both found at the end of the elbow in the shape of the graphs 4.1 and 4.2. The numbers are not far apart, and it could be that the number of neurons in the hidden layer of an ELM trained on Manta Flow data, with which it reaches this point at the end of the elbow in the graph, would also increase to 1000 when more simulations are used to train the ELM. This is because the optimal number of neurons in the hidden layer of an ELM goes hand in hand with the training set size. However, when the same number of training samples are used, ELM's trained on Manta Flow data need less neurons in their hidden layer.

4.2 Number of Samples

4.2.1 Methods

To know how much information is needed for an ELM to be able to train on potential flow and Manta flow data, the number of samples that is used in the ELM's are tested. For both simulations different trials are run with ELM's with different sizes of training sets. Again the height of the sensor plane is at -25 centimeters for the potential flow data, and at -6 frames for the Manta Flow data.

The number of neurons that are added to the hidden layer of the ELM's that will be trained on potential flow data is 1000. In figure 4.1 it can be seen that an ELM with this number of neurons has a MED prediction error of about 13 centimeters. Once we increase the number of neurons further, the MED prediction error gets smaller. We chose not to use a higher number than 1000 neurons because the MED prediction error is sufficient at this number, and training time would be increased if the number of neurons is increased. A higher training time is not desirable because of the computational limitations in this study (section 6.1).

The number of neurons that is added to the hidden layer of the ELM's that will be trained on Manta Flow data is 800. This number is chosen because the MED prediction error in Figure 4.2 is at the same point in the graphs shape at 800 neurons, as the error is at 1000 neurons in 4.1. This means

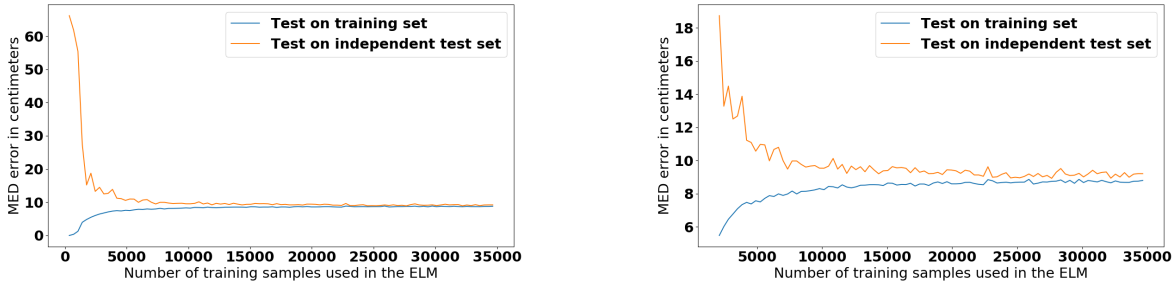

Figure 4.3: The MED prediction error versus the number of samples on which an ELM is trained. The ELM's are trained on potential flow data. In both cases 1000 neurons are used in the hidden layer of the network. The sensor plane is positioned 25 centimeters below the source plane. For the generation of this graph ELM's are trained 100 times, with the number of training samples from 350 to 35000 with an interval of 350 samples. The graph on the left Shows the whole interval, while the graph on the right shows the interval from 1750 to 35000 samples.

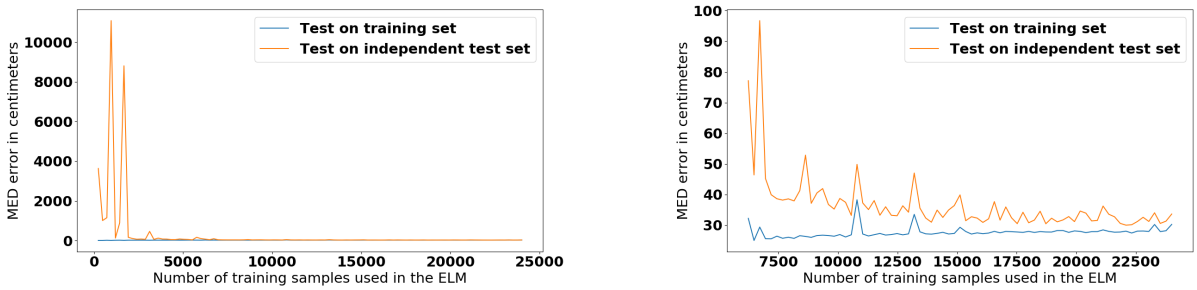

Figure 4.4: The MED prediction error versus the number of samples on which an ELM is trained. The ELM's are trained on Manta Flow data. 800 neurons are added to the hidden layers of the ELMs. The sensor plane is positioned 6 frames below the source plane. For the generation of this graph ELM's are trained 100 times, with the number of training samples from 240 to 24000 with an interval of 240 samples. The graph on the left Shows the whole interval, while the graph on the right shows the interval from 6000 to 24000 samples.

that when training an ELM with these numbers of neurons on the Manta Flow and potential flow data, the ELM'ss are able to reach the same percentage of optimal performance with respect to the hidden layer size and sample size.

4.2.2 Results

The MED prediction error of ELM's trained on potential flow data are shown versus the number of training samples in the two graphs in figure 4.3.

The MED prediction error of ELM's trained on Manta Flow data are shown versus the number of training samples in figure 4.4. In both figures the number of test samples of an ELM is 10 percent of the number of training samples.

For both the ELM's trained on potential flow and Manta flow data, the graphs show a parabolic shape for the MED prediction error of the test set. For potential flow data the asymptote goes to 10 centimeters. The asymptote of the Manta Flow data goes to 30 centimeters. For both simulations the difference in MED prediction error between the test and training set is decreasing when the number of samples in the hidden layer of the ELM's are increasing. At the same number of test samples, there is a bigger difference between the MED prediction error of the test set and the training set for the ELM's trained on Manta Flow data.

4.2.3 Discussion

The ELM's of both simulations reach the minimal MED prediction error around 8000 to 10000 training samples for respectively potential flow and Manta Flow data. In this experiment there are however different hidden layer sizes of the ELM's used for the different simulation methods. The number of neurons in the hidden layer of an ELM and the training set size influence each other. If the hidden layer size is increased, it is likely that an ELM would need more training samples to reach its minimum MED prediction error. Therefore we can conclude that an ELM trained on potential flow data would probably need less training samples if hidden layer sizes would be the same.

When figures 4.3 and 4.4 are compared to each other, it can be seen that the difference in MED prediction error of the ELM's trained on the test set and training set, is bigger at the same training set size for the ELM trained on Manta Flow data. This indicates that there exists a little bit of over fitting which can be prevented by adding more data to the training set. On top of this better results might also be possible if a bigger training set is used together with a higher number of neurons in the hidden layer of an ELM that is trained on Manta Flow simulations.

4.3 Ideal Height

4.3.1 Methods

Because the two simulation methods are very different from each other, the ideal height for the sensor plane to be in in the simulation might also differ. We define the ideal height of the sensor plane as the height with which an ELM can predict objects with the lowest MED prediction error, after training the ELM on that specific height. To test this we use parameters that we can extract out sections 4.1 and 4.2. Again an ELM will be trained multiple times for both simulation methods. The number of neurons that will be used is 1000 for the ELM trained on potential flow data, and 800 for the ELM trained on Manta Flow data. The number of training samples that will be used for the ELM's trained on potential flow and Manta Flow data are both 24000. We stick to the values that we used earlier because these numbers of training samples result in the lowest MED prediction error for the ELM's. These numbers could be lowered because the ELM's seem to have reached their optimum at 7000 and 15000 training samples. Since the number of training samples 24000 is used before, and the increase from 7000 or 15000 to 24000 does not result in a much longer training time, we will again use 24000 training samples in this experiment to be consistent. This choice was made because more samples prevent an ELM from over-fitting.

4.3.2 Results

In figure 4.5 the MED prediction error of an ELM trained on potential flow data are shown versus the height of the sensor plane. Two sets of ELM's were trained for this graph, one where noise was added to the fluid velocities, and one where noise was not added to the fluid velocities. The error of the ELM's that were trained on fluid velocities with noise first

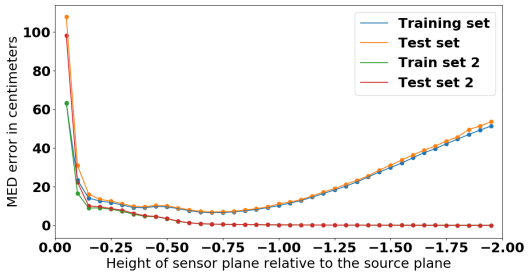

Figure 4.5: MED prediction error versus the height of the sensor plane used in an ELM, while trained on potential flow data. The train set size is 24000, the test set size is 3000. The hidden layer of the network consists of 1000 neurons. The blue and orange lines represent the training and test sets where a Gausian noise of 10^{-6} is added. The green and red lines represent the training and test sets where noise is left out. Two separate ELM's are trained on both the the sets with and without noise.

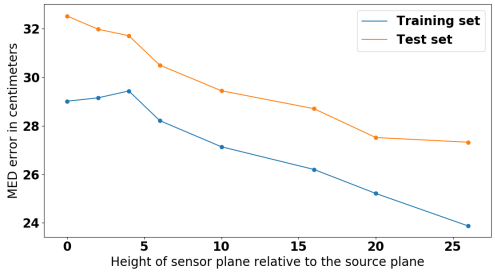

Figure 4.6: MED prediction error versus the height of the sensor plane used in an elm, while trained on data from Manta Flow data. The train set size is 24000, the test set size is 3000. The hidden layer of the network consists of 800 neurons.

declines. After the height of -0.75 the error increases again. For the ELM's that were trained on fluid velocities that did not contain noise, the MED prediction error of the ELM's keeps decreasing, almost reaching zero. In figure 4.6 the MED prediction error of ELM's trained on Manta Flow data are shown versus the height of the sensor plane. The MED prediction error of the ELM seems to be decreasing linearly when the sensor plane gets further away from the source plane. An ELM trained on Manta Flow data cannot be trained on fluid velocities that do not contain noise, since this simulation method itself adds noise to fluid velocities.

4.3.3 Discussion

Because of limited storage in this study, it was decided to store data of sensor plane heights up until 104 centimeters(or 26 frames) below the source plane for Manta Flow simulations. As can be seen in figure 4.6, the MED prediction error of an ELM trained on Manta Flow data keeps on decreasing until this point. It is likely that we therefore did not find an optimal sensor height for the ELM trained on Manta Flow data because this sensor height is probably further away tan 104 centimeters.

The ideal height of the sensor plane for ELM's trained on Manta Flow data is lower than the ideal height of the sensor plane for ELM's trained on potential flow data. A height that is too close to the source plane seems to have a negative effect on the MED prediction error of ELM's trained on both simulation methods. It looks as if the viscosity that is included in Manta Flow simulations, causes the fluid velocity to be especially noisy in the beginning. By this we mean that there are bigger differences in the individual fluid velocities at grid points, where the differences look like randomness. This effect seems to decrease once a sensor is placed further away from the source that is causing the fluid flows.

5 Sensor Positions

The final subject that is investigated is the optimal sensor configuration in a potential flow and Manta Flow simulation, with which an ELM can predict the source location of an object with a minimal MED prediction error. This is determined by a genetic algorithm. Because the genetic algorithm is a very computational intensive program which takes a very long time to run, concessions are needed between performance and speed. Parameters are set so that performance is almost at its optimum, but speed is also increased. The genetic algorithm is

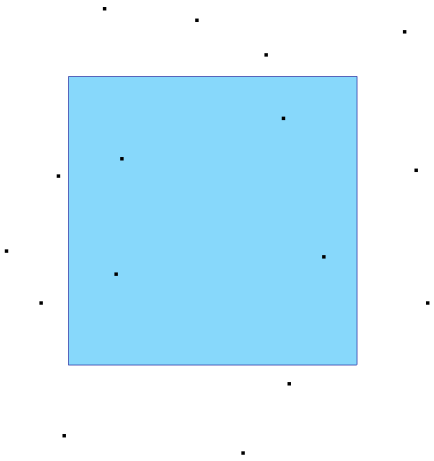

Figure 5.1: Sensor plane of 4 by 4 meters.

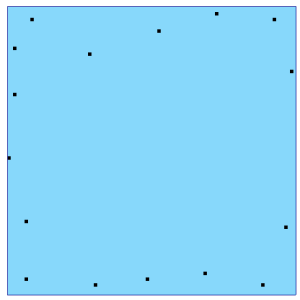

Figure 5.2: Sensor plane of 2 by 2 meters.

The sensor positions that had the lowest MED prediction error after the last iteration of the genetic algorithm, which used potential flow data. The blue square represents the object plane, the black dots represent the sensors that are placed 45 centimeters below the object plane.

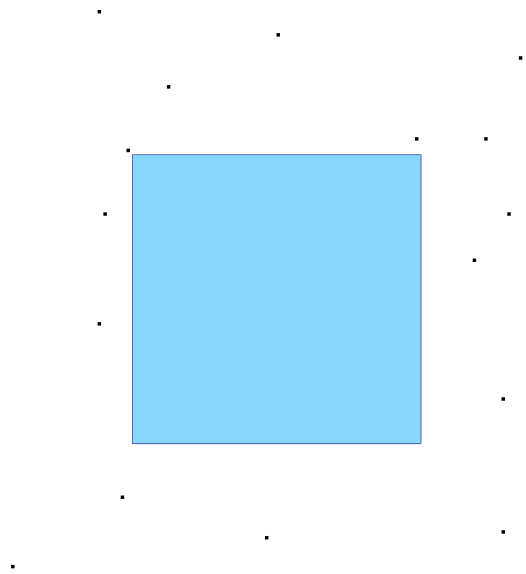

Figure 5.3: Sensor plane of 100 by 100 grid points.

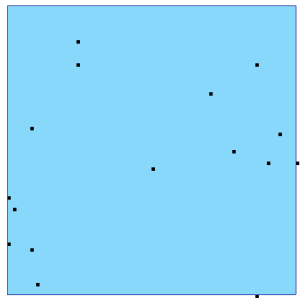

Figure 5.4: Sensor plane of 50 by 50 grid points

The sensor positions that had the lowest MED prediction error after the last iteration of the genetic algorithm, which used Manta Flow data. The blue square represents the object plane, the black dots represent the sensors that are placed 26 frames points below the object plane.

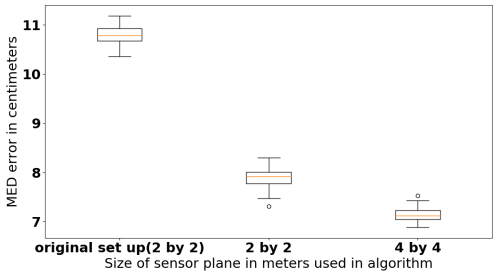

Figure 5.5: Elm's trained on potential flow data. Sensor positions are found in figures 3.2, 5.2 and 5.1.

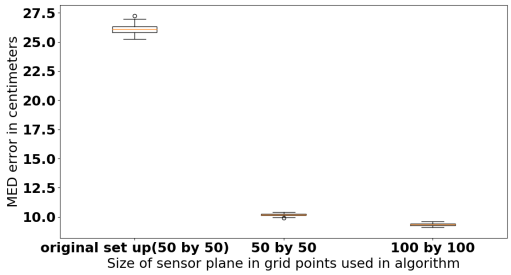

Figure 5.6: ELM's trained on Manta Flow data. Sensor positions are found in figures 3.2, 5.4 and 5.3.

These box plots show the MED prediction errors of ELM's that are trained on fluid velocities at sensor positions found in the basic sensor set up, and the optimal sensor positions found by the genetic algorithm. For each set of positions, an ELM is trained 50 times.

run twice with different sensor plane sizes for both simulation methods.

5.1 Initial Population

The algorithm starts with an initial population of 50 individuals. These individuals have 16 sensor positions witch each an x and y position as their genes. The positions of the sensors can be in a 2 by 2, and a 4 by 4 meters grid, 45 centimeters below the 2 by 2 meters source plane in the case of potential flow. In the case of Manta Flow the sensors can be in a 50 by 50, and a100 by 100 points grid, 26 frames below the 50 by 50 points source grid. The x and y positions of the sensors of each individual are generated semi-random. The sensor plane is first divided into 16 equal squares that do not overlap and do cover the whole plane. Then each sensor gets a random x and y position in the form of an integer in one of the squares. Each square can be used only once. In this way we space out the sensor positions more equally than with a totally random initialization of the sensor positions. In this way the computation time of the genetic algorithm until an optimum is reached is sped up, because less iterations of the genetic algorithm are needed.

5.2 Selection

After the initialization of the population, the top ten individuals that have the lowest MED prediction error are chosen to survive. This selection is done by the algorithms fitness function. The fitness function takes all sensor position of an individual, and uses them to train an ELM on the fluid velocities at those positions. We use respectively 7000 and 15000 training samples, and 1000 and 800 neurons in the hidden layers, for the ELM's trained on respectively potential flow and Manta Flow data. These numbers seem to be the minimum for the MED prediction error to be almost at its lowest point if we look at Figures 4.3, 4.4, 4.1 and 4.2. The score that the fitness function then assigns to each individual is the MED prediction error.

5.3 Generation

The next step is to generate a new population with the selected top individuals. First each individual makes a new child with every other individual. It is randomly decided for each of the 16 sensor positions which position is given to the child from either one of the parents. After the initiation of the new children there is a 10 percent chance of a mutation of each x and y coordinate separately, of each sensory position of each child. When mutated, a coordinate has a uniform random chance to shift between -50 and 50 percent of the total width of the sensor plane. This is a relatively high mutation rate for a genetic algorithm. This rate is chosen to make up for the fact that only 50 individuals are used in each population. With this high rate the chances are increased to find a better solution. To

make sure that this mutation rate does not impair the performance of the algorithm by mutating already good individuals, the top 5 individuals of the last generations are also one on one copied into the new generation. A total of $5+\sum_{n=1}^{10-1} n = 50$ new individuals are created for the new population. This population is then used in the next iteration of the algorithm and another selection takes place. Important to notice is that also during the generation of the new population the sensor position coordinates remain integers.

5.4 Termination

The process of selection and generation is repeated 50 times after which the optimal sensor positions for both simulation methods are chosen. The optimal sensor configuration is the set of sensor positions of the last iteration of the algorithm, that has the lowest MED prediction error.

5.5 Results

Figures 5.1 and 5.3 show the sensor positions which were assigned the lowest, and thus the best score by the fitness function of the genetic algorithm after the last iteration, for respectively data of the potential flow and Manta Flow simulations. The blue squares in these figures are the 4 by 4 meter, or the 100 by 100 grid points planes in which objects are placed in the simulations. The sensors are indicated by the black dots. The sensor positions that are best according to our genetic algorithm in both cases, are mostly outside of the plane where objects are in.

Figures 5.2 and 5.4 show the best sensor positions according to our algorithm when we limit the sensor plane to be the same size as the source plane. In the four figures mentioned in this section, it can be seen that all sensors are pushed to the outside. In the figures where the sensor plane was bigger than the source plane, almost all sensors would lie outside of the source plane.

6 Discussion

6.1 Computational and Storage Limitations

For this study there was a limited amount of storage, and a limited amount of computational power available. Because of this, some settings were down scaled at the cost of performance. The Manta Flow simulations have only a resolution of 100 by 100 grid points, which corresponds in our simulation to 4 by 4 meters. Because this simulation was in 3d, 45 frames were needed on top of each other. A round object that was instantiated in this grid would show some significant edges because of the low resolution. When a higher resolution is possible, lower MED prediction errors of ELM's might be achieved, due to a more smooth object.

6.2 Fitness Function Genetic Algorithm

The fitness function of the genetic algorithm used in this study uses an ELM, and calculates the MED prediction error of the ELM. The fact that the fitness function uses an ELM, together with the Gaussian noise that is added to the fluid velocities of the simulation, causes the score of the fitness function to involve a bit of chance. If an ELM is trained twice on the same training set, the MED prediction error can differ a little bit. This results in the fact that our genetic algorithm might not have found the most optimal sensor configuration. However, because the best sensor position sets are one on one given to each next generation, it is very much likely for the most optimal solution to have survived until the last iteration of the algorithm. When working with a fitness function that does not assign a purely mathematical score to each individual in the population, it is not hundred percent certain that the optimal sensor positions found are indeed the optimal positions.

On top of that, the population size that could not be made too big because of the storage and computational limitations of this study. It was chosen to have a population size of 50 individuals. The genes of these individuals are chosen randomly and should have a sufficient, but not very high variety in the beginning. The high mutation rate that the

algorithm tries to make up for the fact that our variety is not very big. The mutation rate does however not retain the performance of the algorithm because the best solutions are also passed on to the next generation. For this study it was tried to use the most optimal settings for the computational power that was available, but results might be show lower MED prediction errors of ELMs when more computational power is available so that a higher population can be chosen.

6.3 Manta Flow vs. Potential Flow

The two simulation methods that are used in this study are fundamentally different. The potential flow simulation is a purely mathematical approach which uses a few vectors as an input, and gives a fluid velocity flow vector as an output at a certain point. Manta Flow is a simulation method where for each time step the new values are being calculated. Viscosity is taken into account in this method. Manta Flow does however not use the metric system. Instead it uses grid points. In this study we tried to simulate in a way that 1 grid point would equal 4 centimeters, based on the paper of Wester(2019).

6.4 Number of Neurons vs. Training Set Size

There exists a relation between the number of neurons that are added to a hidden layer of an ELM and the training set size that is used. As can be seen in figures 4.1 and 4.2, as we increase our number of neurons, the difference in MED prediction error when tested on the training set and test set is increased. When this difference gets higher, the network is more likely to suffer from over fitting. It might be interesting in another study to extract a mathematical relation between the number of available training samples, and the maximum size of the hidden layer with which the ELM does not fall victim to over fitting.

7 Conclusions

7.1 Parameters

In section 4, different parameters of an ELM are tested to find the settings with which an ELM has the lowest MED prediction error. ELM's trained on Manta Flow data need a smaller hidden layer size to reach the lowest MED prediction error compared to ELM's trained on potential flow data. ELM's in both cases need the same number of samples in their training set. This is however with different numbers of neurons in their hidden layers, because the numbers that are used are the number of neurons with which both ELM's had reached a certain percentage of their lowest MED prediction error. The height of the sensor plane that results in the lowest MED prediction error of an ELM is lower when Manta FLow data is used, than when potential flow data is used.

7.2 Genetic algorithm: sensor positions

The optimal sensor positions as determined by the genetic algorithm for both simulation methods, when the sensors were not restricted to be in a plane as big as the source plane, show similar results. Most sensors are being place outside of the source plane, and are evenly spread out across the area outside of the source plane. The optimal sensor positions when the sensor plane is limited to the area of the source plane, is different for both methods. Where the sensors in the potential flow simulations are all pushed to a distance only a few centimeters from the sides, the sensors in Manta flow simulations are more spread out throughout the field. These sensors seem to be pushed a bit to the outside of this plane as well, but not as much. The difference here might be an adaptation to handle viscosity in the fluid flows better by placing the sensors at different places. Further research must however confirm if this sensor positions are still resulting in the lowest MED prediction error with a bigger training set of the ELM's used in the genetic algorithm, a bigger population used in the genetic algorithm, and a higher resolution used in the Manta Flow simulations.

Another interesting finding is that the MED prediction error of an ELM that used Manta Flow data can be lowered by about 70 percent when the optimal sensor positions displayed in figure 5.3 are used, compared to the error when the basic sensory set up is used(figure 3.1). The MED prediction error of an ELM that used potential flow data can only be lowered by about 35 percent when the optimal

sensor positions displayed in figure 5.1 are used, compared to the MED prediction error when the basic sensory set up is used. This means that the points where sensors are placed to measure fluid flows probably have a bigger influence when using viscous fluid flow simulations. This however also needs to be confirmed by future research where a higher simulation resolution, bigger data set, and bigger population in the genetic algorithm are used.

Concluding, it seems to be the case that ELM's trained on fluid velocities that include viscosity, need a bigger training set because the viscosity makes fluid flows appear to be more random. To measure the fluid velocities at a greater distance from the objects source, has a bigger decline in MED prediction errors for ELM's trained on simulations where viscosity is taken into account, even though it also has a decline in MED prediction error on the ELM's trained on simulations where viscosity is not taken into account. On top of this the ELM's which used viscous fluid flows seem to be more sensitive to the optimal sensor positions for measurements because the MED prediction error can be decreased twice as much in relative terms compared to ELM's which used non-viscous fluid flows.

References

S.M. van Netten B. Curcic-Blake. Source location encoding in the fish lateral line canal. *Journal of Experimental Biology v209 n8: 1548-1559*, 2006.

Luuk H Boulogne, Ben J Wolf, Marco A Wiering, and Sietse M van Netten. Performance of neural networks for localizing moving objects with an artificial lateral line. *Bioinspiration & Biomimetics*, 12(5):056009, sep 2017. doi: 10.1088/1748-3190/aa7fcb.

S. Dijkgraaf. The functioning and significance of the lateral-line organs. *Biological Reviews v38 n1: 51-105*, 1963.

Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, volume 2, pages 985–990 vol.2, July 2004. doi: 10.1109/IJCNN.2004.1380068.

John H. Holland. Outline for a logical theory of adaptive systems. *J. ACM*, 9(3): 297–314, July 1962. ISSN 0004-5411. doi: 10.1145/321127.321128.

Jacob Engelmann Sophia Kröther Horst Bleckmann, Joachim Mogdans and Wolf Hanken. Wie fische wasser fuhlen: das seitenliniensystem. *Biologie in unserer Zeit, 34(6):358365*, 2004.

Runwei Cheng Mitsuo Gen. *Genetic Algorithms and Engineering Design*. John Wiley, 1997. ISBN 9780471315315.

Primoz Pirih and Sietse M van Netten. Theory of hydrodynamic imaging: Potential flow in three dimensions. *Part of a deliverable for the Lakhsmi Project*, 2018.

C. Steenkist. Comparing and reducing dark corners in lateral line sensor configurations. Bachelor's thesis, Rijksuniversiteit Groningen, 2019.

Nils Thuerey and Tobias Pfaff. MantaFlow, 2018. *http://mantaflow.com*.

Andreas N. Vollmayr, Stefan Sosnowski, Sebastian Urban, Sandra Hirche, and J. Leo van Hemmen. *Snookie: An Autonomous Underwater Vehicle with Artificial Lateral-Line System*, pages 521–562. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. ISBN 978-3-642-41446-6. doi: 10.1007/978-3-642-41446-6_20.

C. Wester. Comparing and reducing dark corners in lateral line sensor configurations. Bachelor's thesis, Rijksuniversiteit Groningen, 2019.

B. Wolf and S. van Netten. Training submerged source detection for a 2d fluid flow sensor array with extreme learning machines. In *International Conference on Machine Vision (ICMV 2018), At Munich, Germany*, volume 11, 2018.