



# COMPARING AND REDUCING DARK CORNERS IN ARTIFICIAL LATERAL LINE SENSOR CONFIGURATIONS

Bachelor's Project Thesis

Christiaan Steenkist, s2744244, c.n.steenkist@student.rug.nl

Supervisors: S. M. van Netten, P. Pirih and B. J. Wolf

**Abstract:** Fish are able to sense movement in the water around them with an organ called a lateral line. This organ consists of neuromast detectors along the body of the fish that detect water flow. For an artificial lateral line, artificial water flow sensors are used. Fisher matrix heatmaps of these bi-axial sensors show that there are areas where parameter detection accuracy is low. The goal of this paper is to reduce these dark corners for the position of the source of the flow. The assumption is that regularly spaced out sensors, such as in a grid configuration, have the least dark corners. Sensor configurations generated by a Continuous Genetic Algorithm, K-means, Kohonen SOM and a grid configuration were compared. Extreme Learning Machines were trained and tested on a dataset of moving sources and compared with dark corners from Fisher matrix heatmaps. The parameter errors of the ELM are approximately 4 times larger than the lower bounds of the Fisher matrices. The dark corners shown by Fisher matrix heatmaps, however, are not visible in the ELM parameter errors. The CGA configurations had the highest ELM errors, except for an outlier, and had the most dark corners. The grid configuration had the least dark corners and shared the lowest ELM errors with the sensor configurations generated by K-means and Kohonen SOM.

## 1 Introduction

Fish can sense fish or other moving objects in the water around them. They do this with a sensing organ called the lateral line. Lateral lines contain sensing organelles called neuromasts that react to water flow. Fish use this sense for tasks such as detecting prey (Hoekstra and Janssen, 1985) and schooling with other fish (Partridge and Pitcher, 1980).

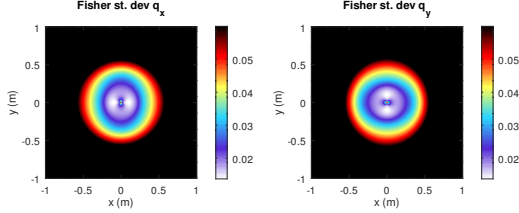
Being able to know the position and motion of underwater objects can be very important. Submarines and other submersibles can have their sensory capabilities upgraded by employing artificial lateral lines. These would supplement sonar, radar and touch sensors. Collections of artificial lateral lines can be used to keep track of the traffic in ports and therefore be used to prevent collisions. The migration of fish along a river can also be recorded.

An artificial lateral line was created in the lab and was able to detect and localize the vibrating tail of a crayfish (Yang et al., 2010). The artificial

neuromasts, like the neuromasts of the fish, produce electrical signals when bent by water flow. In this lateral line, mono-axial sensors in two orientations were used.

These mono-axial, but also bi-axial sensors, can be analyzed to reveal the capabilities and limits of the sensors. It is useful to know what sources can and cannot be detected. One such way is by using Fisher Information.

Fisher information matrices for artificial lateral line sensors are calculated with a model for the detected water flow. The model calculates the sensor reading from a number of source parameters and the sensor position. For artificial lateral line sensors these source parameters are the position of the moving source, the source velocity vector and the source radius. The problem is reconstructing the values of these parameters from sensor readings. Fisher information provides a theoretical limit on the accuracy of the reconstruction and prediction. These limits are separate for the different source parameters and components of these parameters.



**Figure 1.1: Source position ( $q_x, q_y$ ) Cramér-Rao lower bounds 25 centimeters above a single bi-axial sensor. The heatmap grid points are calculated with smaller Fisher matrices. There are no rows or columns for the  $z$  components or the source velocity vector. For single sensors the dropoff is radial with a dark corner above the center of the sensor.**

This is called the Cramér-Rao lower bound which is an unbiased lower bound of the parameter estimation error.

The individual values in the Fisher information matrix are calculated from the partial derivatives of the water flow model. The Cramér-Rao bound varies for different source positions, movements and sizes as a consequence. The Cramér-Rao bounds of various locations around mono-axial, biaxial and tri-axial sensors were looked at with the potential flow model as the water flow model (Pirih and van Netten, 2018). This led to the discovery of dark corners surrounding the sensors in single and multi-sensor configurations. In these dark corners, one or more of the position parameters or velocity parameters can not be measured accurately.

In rivers and ports the position and movement along the  $z$ -axis may not be important. If the number of parameters is reduced then the number of rows and columns in the Fisher matrices are also reduced. This leads to different Cramér-Rao bounds and so the dark corners change. Figure 1.1 shows the errors around a sensor as calculated with reduced Fisher matrices. The error increases mostly radially from the sensor and only marginally directly above.

Aside from the Fisher information there is a different way to look at the performance of artificial lateral line configurations and dark corners. There has been some success with Extreme Learning Machines (ELM) to predict source position and velocity vectors (Boulogne et al., 2017). The Mean Squared Error (MSE) and Mean Euclidean Distance (MED) are used as measures of the performance of the ELM. Fisher information only gives a lower bound on the standard deviation of the source parameters. Comparing ELM performance with the Cramér-Rao lower bound can therefore give a more realistic view of ALL capabilities. Dark corners can also be found in the ELM predictions and be compared to earlier findings.

The main goal in this paper is to compare the performance of the ELM to the predicted optimum of the Fisher matrices. This is done by comparing the Cramér-Rao lower bound of various locations to the ELM error at those locations. The focus lies on the source position.

The second goal is to reduce the amount of dark corners. This is done by varying the positions of the sensors in the sensor configuration. The configurations will be compared with a fitness score that has its basis in Fisher Information. The ELM Mean Squared Errors and Mean Euclidean Distance are also used. The configurations with the least dark corners have then the lowest values of these metrics.

## 2 Background

Several methods from previous research into Artificial Lateral Lines have been used. The Potential Flow model was used as model of the water flow that is sensed by the sensors. Fisher Information Matrices were used to analyze the area around sensors and sensor configurations. Extreme Learning Machines were used to test the performance of the sensor configurations with respect to a dataset of moving sources.

### 2.1 Potential Flow

The water flow is calculated with the sensor position  $\vec{s}$  and source position  $\vec{q}$  vectors. The potential flow equation simplifies to equation 2.1 by placing the sensor at the origin (Pirih and van Netten,

2018). Here  $\vec{r}$  is the difference in position between the sensor and source  $\vec{r} = \vec{q} - \vec{s}$ . The other parameters in this equation are the source radius  $a = 0.06$  meters and the source velocity vector  $\vec{w}$ . The resulting vector  $\vec{v}$  is the waterflow as observed at the sensor position.

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

## 2.2 Fisher information

Fisher information matrices are used to calculate the Cramér-Rao bound. This is an unbiased lower bound on the prediction error of the parameters of a model. For this and previous research, the model is the Potential Flow model of water flow.

Partial derivatives of the Potential Flow model are used to calculate the values of the Fisher matrices. These are the partial derivatives over the water flow speed  $\vec{v}$ . The Fisher information matrix and thus the partial derivatives are calculated for a set of Potential Flow parameters. The Fisher matrix for a source location, sensor location and a source velocity vector is calculated with a list of partial derivatives  $D$  as shown in equation 2.2. Calculating the matrix  $F$  is shown in equation 2.3 which creates a square matrix. Here  $\sigma$  signifies the standard deviation of gaussian sensor noise.

Inversing the square matrix yields the covariance matrix for the parameters corresponding to those of the partial derivatives. Combining multiple sensors together can be done by calculating matrices for all sensors  $s_1$  through  $s_n$  and summing them before inverting into the covariance matrix. This is shown in equation 2.4 where  $i$  is the sensor number.

$$\vec{D}_i = (\frac{\partial v_i}{\partial q_x}, \frac{\partial v_i}{\partial q_y}) \quad (2.2)$$

$$\mathbf{F}_i = \frac{1}{\sigma_{v_i}^2} * \vec{D}_i^T * \vec{D}_i \quad (2.3)$$

$$\mathbf{C} = (\sum_{i=1}^n \mathbf{F}_i)^{-1} \quad (2.4)$$

### 2.2.1 Fisher heatmap

The partial derivatives have to be filled in to calculate the Fisher matrix. The heatmap points are

used as source positions when creating a Cramér-Rao heatmap. For  $w$ , Fisher matrices are calculated for  $m$  velocities  $w_1$  through  $w_m$  and averaged. To represent all directions equally the positive and negative directions along the  $x$  and  $y$  axes as well as the normalized diagonals between them are used. This gives a total of eight directions. These are scaled with a velocity of 0.13 meters per second which is taken over from the previous ELM work (Boulogne et al., 2017). The 0.06 meters source radius is also taken from the same source.

The matrices from all velocities are first averaged into one matrix per heatmap point per sensor. The matrices are then summed over all sensors so only one matrix remains per point. All matrices are then inverted yielding the parameter covariance matrices. The diagonals are extracted and the square roots taken so that only the standard deviations ( $\sigma_{q_x}, \sigma_{q_y}$ ) remain.

## 2.3 Extreme Learning Machines

The ELM is a simplified neural network related to the Echo State Network. Instead of a pool of neurons the ELM has a hidden layer with random weights to the input layer. The weights to the output layer are directly calculated from the training set in a single batch which makes training quick compared to other algorithms.

### 2.3.1 ELM training

Each ELM is trained using a fixed dataset of sources that are transformed through a sensor configuration into water flow data according to section 2.1. For this investigation, 16 bi-axial sensors are used. Each sensor detects water flow in the directions corresponding to the  $x$  and  $y$  axis of the pool. These water flow components are the sensor outputs leading to 32 outputs for the 16 sensors. The 32 total sensor outputs and 1 bias mean the ELM has 33 inputs. The hidden layer size is 5000 nodes (taken from Boulogne et al. (2017)). The same hidden layer weight matrix is used for every ELM.

The sensor output for each source is linearly scaled after the sensor noise has been added. After scaling, the absolute value of the highest sensor output is 1 with the other outputs scaled appropriately. The sensor data of all samples with a bias appended as 33rd input  $u$  is first multiplied with

the hidden layer weights  $W_h$ . This output is transformed with the activation function  $\tanh$  to limit the output of the hidden layer layer (eq. 2.5). The next step is to multiply the output data  $d$  of the training set with the Penrose-Moore pseudoinverse ( $M^\dagger$ ) (eq. 2.6). Not pictured is the bias row that is appended to  $M$  as input 5001 between equation 2.5 and 2.6.

$$M = \tanh(u * W_h) \quad (2.5)$$

$$W_{out} = d * M^\dagger \quad (2.6)$$

### 2.3.2 ELM testing

Predictions by the ELM can be made when the hidden layer weights  $W_h$  and the output layer weights  $W_{out}$  are known. These predictions are made for a single source at a time. The testing uses the same scaled sensor input as the training. The intermediate output  $M$  is calculated in the same manner as during training (section 2.3.1) with a bias row added to the sensor data and to  $M$ . To get the parameter prediction  $p$  the intermediate output  $M$  is multiplied with the weights of the output layer as shown in equation 2.6.

$$p = M * W_{out} \quad (2.7)$$

## 3 Methods

### 3.1 Sensor configuration testing

There are two ways by which a configuration can be analysed and scored. An error heatmap for source parameters can be made by calculating and inverting a Fisher matrix for each heatmap point. This gives the Cramér-Rao lower bound on error of the model parameters. The second way is by letting an extreme learning machine train on a dataset of source location. This dataset is transformed by the sensor configuration and the flow model into sensor output. The errors of the predictions of the ELM give an indication of the fit of the configuration to the data given that the ELM is able to predict the data.

For both scoring methods an underlying model is needed. The Potential Flow model was chosen because ELMs are already able to predict source loca-

tion using this model. It has also been found to give a good approximation of the responses by neuro-masts in fish caused by oscillating spheres (Ćurčić-Blake and van Netten, 2006). This is a valid model for our experiment as the source velocity is not that large compared to the source radius.

#### 3.1.1 Source parameters

Equation 2.1 shows that the source velocity vector  $\vec{w}$  and source radius cubed  $a^3$  are directly multiplied with each other. A large source moving slowly may generate the same flow pattern using potential flow as a small source moving more quickly. When the source radius is fixed, the true source velocity  $\vec{w}$  can be extracted from the sensor data. The remaining source parameters are the three location components ( $q_x, q_y, q_z$ ) and the three movement components ( $w_x, w_y, w_z$ ). This is further reduced to ( $q_x, q_y, w_x, w_y$ ) as the  $z$  components are not predicted and so dropped. While  $q_z$  and  $w_z$  are not predicted, they are still used in the sensor model. The velocity component  $w_z$  is always 0 and the source height  $q_z$  was arbitrarily chosen to be 0.25 meters. When the velocity components are dropped the model parameters become ( $q_x, q_y$ ).

### 3.2 Fisher Fitness Function

The Continuous Genetic Algorithm (section 3.3.1) requires a fitness function to order the sensor configurations. To create this fitness function, the variances of the parameters from a Fisher heatmap are combined into a single score. This way we can assign scores to sensor configurations.

In equation 3.1 the parameter variances are combined into a score. The score is then averaged over all heatmap points into the Fisher Fitness Function for the sensor configuration. This is shown in equation 3.2 where  $\langle \rangle$  indicate the heatmap grid average.

$$F_{score}^2 = \sigma_{q_x}^2 + \sigma_{q_y}^2 \quad (3.1)$$

$$FFF = \langle \sqrt{\sigma_{q_x}^2 + \sigma_{q_y}^2} \rangle \quad (3.2)$$

#### 3.2.1 Source dataset

The locations of the sources are uniformly distributed over the range  $[-1, 1]$  for both  $x$  and  $y$

coordinates. The source velocity vectors have a uniformly distributed angle over the range of  $\phi = [0, 2\pi]$ . The velocity of the movement is fixed at 0.13 meters per second.

### 3.2.2 ELM metric

The ELM errors, Mean Squared Errors (MSE) and Mean Euclidean Distance (MED) should give a more realistic evaluation of a configuration. The results that the Fisher matrices provide are optimal results and may not be attainable. The ELM results should be a closer approximation of real life performance compared to the Fisher information.

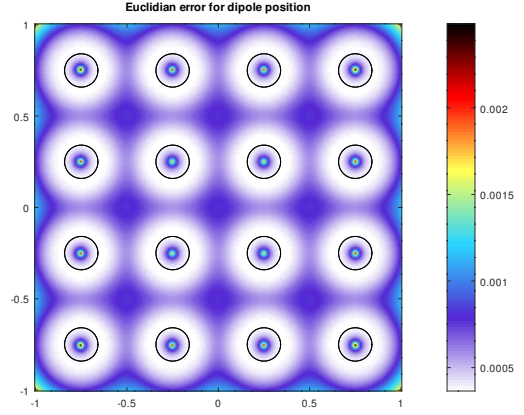
## 3.3 Configuration generation

In the interest of reducing dark corners, several configurations can be compared. To generate an optimal sensor configuration of 16 sensors, a Continuous Genetic Algorithm (CGA) (Chelouah and Siarry, 2000) was trained with the Fisher Fitness Function. Sensor configurations as generated by K-means (MacQueen, 1967) and Kohonen Self Organizing Map (Rougier and Boniface, 2011) are also used. Both K-means and Kohonen SOM were chosen because of the radial accuracy falloff as indicated by the Fisher heatmaps. A grid configuration (figure 3.1) is used as a reference.

All the sensor configurations are bound to a 2 by 2 meter square sensor plane. The moving sources are positioned on an identical square source plane that lies 25 centimeters above the sensor plane. These distances are arbitrary. However, they fit the source velocities and size that were used and thus the corresponding water flow and noise level.

### 3.3.1 Continuous Genetic Algorithm

The CGA performs like a normal genetic algorithm but for continuous instead of discrete parameters. This requires that the crossover and mutation methods be changed (Chelouah and Siarry, 2000). Besides these two methods the algorithm was implemented as usual. The parameters that the CGA will try to optimize are the  $x$  and  $y$  position of the sensors. For 16 sensor configurations this means that the dna of each individual configuration contains 32 floating point numbers.



**Figure 3.1: A euclidean error heatmap 25 centimeters above the fixed point of reference configuration. The bi-axial sensors are indicated by the black circles. Note the dark corners directly above each sensor.**

The fitness function for the CGA is the Fisher metric. To calculate the metric, a 51 by 51 grid over the source plane is used.

### 3.3.2 The k-means clustering algorithm

The K-means algorithm is a well-known clustering algorithm. The algorithm clusters groups into  $K$  sets. (MacQueen, 1967) It does this by assigning samples to centroids or neighbourhoods.

First all samples are assigned to their closest centroid. The centroids are then moved to the mean of all their assigned samples. This process repeats until the movement stops or some predetermined threshold is reached.

To generate our configurations the algorithm stops when the Fisher metric, with the centroids as bi-axial sensors, stops decreasing. This is calculated along a 201 by 201 grid over the source plane. This calculation is done once every 10 cycles.

The data set that the algorithm runs on, contains the positions of the sources that are used for the fisher Heatmap. This means that the uniform grid of 201 by 201 samples over the source plane is used. At the start of the algorithm, the centroids are randomly distributed over this space.

### 3.3.3 The Kohonen self organizing map

The Kohonen self organizing map (Kohonen SOM) is a linked neural map that forms itself to a large number of presented vectors. (Rougier and Boniface, 2011) For this experiment the vectors come from a uniform distribution of positions on the source plane. The neural map is made out of  $n$  codewords, equal to the number of sensors. When the codewords are replaced by sensors the algorithm essentially fit the sensors so they encompass as much of the pool area as possible.

**Kohonen SOM training** Each vector is assigned to a winning code  $b_g$  in the neural map. This is the code that has the smallest distance to the vector  $c$ . The winning code as well as the other codewords are then shifted towards the vector  $b$  depending on their distance to the winning codeword, the current learning rate and the neighbourhood size.

With every vector the learning rate and neighbourhood size decrease to reduce the changes over time. This can be seen in equations 3.3 and 3.4 where  $t$  is the number of the current sample from 0 for the first up to  $t_f$  for the last sample. Here  $\tau_i$  and  $\varepsilon_i$  are the initial neighbourhood size and learning rate and  $\tau_f$  and  $\varepsilon_f$  are the final size and rate.

This distance between the codes is normally given by the connections between the codes that make up the map lattice. Instead of using a lattice and lattice distance the euclidean distance between the codes is used. As a result of this the starting and final neighbourhood sizes are required to be smaller than the one used by Rougier and Boniface in their similar test with a uniform square distribution. The formula for the moving of the codes can be seen in equations 3.5 and 3.6. At the end of training the code positions are taken as sensors and the codebook becomes the sensor configuration.

$$\tau(t) = \tau_i * \left(\frac{\tau_f}{\tau_i}\right)^{t/t_f} \quad (3.3)$$

$$\varepsilon(t) = \varepsilon_i * \left(\frac{\varepsilon_f}{\varepsilon_i}\right)^{t/t_f} \quad (3.4)$$

$$\Delta b_i = \varepsilon(t) h_\tau(t, i, g)(c - b_i) \quad (3.5)$$

$$h_\tau(t, i, g) = e^{-\frac{|b_g - b_i|^2}{2\tau(t)^2}} \quad (3.6)$$

## 3.4 Experiment parameters

### 3.4.1 ELM Parameters

The source dataset was discussed in section 3.2.1 The training set contains 24000 samples to guarantee good training. The testing set, in comparison, is only 3000 samples. The hidden layer size is 5000 as mentioned in section 2.3.1. The hidden layer weights are fixed for every ELM. A gaussian error ( $\mu = 0, \sigma = 1 * 10^{-6}$ ) is added as sensor noise (in  $m/s$ ) to each water flow component of each bi-axial sensor.

### 3.4.2 Cramér-Rao heatmaps

The Cramér-Rao error bound heatmaps are made with a 201 by 201 uniform grid over the source plane. The amount of gridpoints can be increased or decreased for higher resolution or faster computation time, respectively. The movements used are the directions along the  $x$  and  $y$  axis as well as the normalized diagonals between them scaled to 0.13 meters per second. The source radius used is 0.06 meters. The sensor noise used in the heatmaps is the same gaussian noise that is used in the ELM.

### 3.4.3 CGA parameters

There were 30 CGA runs with 100 cycles each. Each run the population went linearly from 40 individuals down to 10 over the course of the cycles (rounded up to the nearest multiple of 2). The crossover rate was 75% and the mutation rate was 1%. The mutation rate went down linearly to 0% over the course of the cycles. The fitness function is to calculate the Fisher metric with a 51 by 51 uniform grid over the sensor area but otherwise standard parameters.

### 3.4.4 Kohonen SOM and k-means

Both Kohonen SOM and K-means generated a configuration 100 times. K-means configurations are generated according to section 3.3.2 on the same 201 by 201 uniform grid that is used by most Fisher matrix calculations. Kohonen SOM configurations are generated by training on 100000 randomly picked vectors.

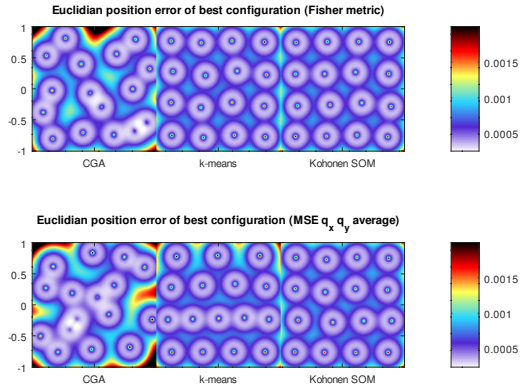


Figure 4.1: A visualization of the best sensor configurations. The top three configurations are the best of the CGA, K-means and Kohonen SOM methods when scored with the Fisher Fitness Function. The bottom three configurations are the best of those methods but according to the ELM average Mean Squared Error of the  $q_x$  and  $q_y$  parameters.

## 4 Results

### 4.1 Best Configurations

With the three methods (section 3.3) a number of sensor configurations were generated. The ones with the lowest errors according to the Fisher Fitness Function can be seen in the top row of figure 4.1. The best configurations according to the ELM errors can be seen in the bottom row of the same figure. The scoring method used here was the average of the  $q_x$  and  $q_y$  Mean Squared Error.

The sensor configurations that were generated by the CGA, are the most varied. These have therefore been used to directly compare the Fisher matrices and ELM.

### 4.2 ELM compared to Cramér-Rao

The errors from testing with an ELM can be compared to the Cramér-Rao lower bounds that were calculated with Fisher matrices. The ELM errors are all relatively normal (figure 4.2). This means we can take the standard deviation of the  $q_x$  and  $q_y$  errors. For the Cramér-Rao lower bound we can use the Fisher heatmaps. These can be averaged

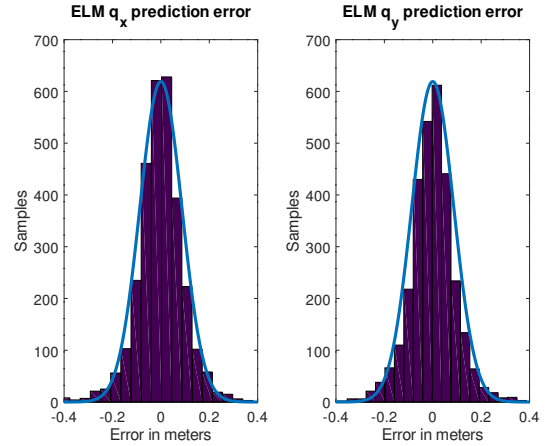


Figure 4.2: The position ( $q_x, q_y$ ) ELM errors of one of the CGA configurations. The errors are normally distributed around 0. The blue envelope shows a normal distribution that fits the errors.

over the entire pool to yield the average standard deviation for each parameter.

This comparison was made with the 30 CGA configurations (figure 4.3). The ELM is neatly separated from the average lower bound. According to the Fisher information, the difference between the different CGA configurations should be small. On the other hand, the ELM sees a larger difference between configurations. The standard deviations of the ELM error are approximately 4 to 5 times larger than the Cramér-Rao lower bound.

#### 4.2.1 Other configurations

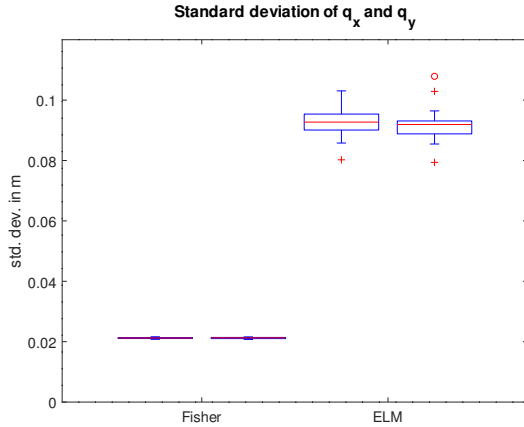
The best configurations (section 4.1) were also compared using this method (table 4.1). There is not much difference between the  $\sigma$  that were calculated with Fisher information. The largest difference between the sigmas of different configurations is a single millimeter. The largest difference between the ELM sigmas of the different configurations is more than a centimeter. The difference between Cramér-Rao lower bound and ELM performance is again a factor 4 to 5.

### 4.3 Comparing heatmaps

It is also possible to create heatmaps from the ELM errors. For this the best CGA configuration ac-

**Table 4.1: The standard deviations of the errors of source position ( $q_x$  and  $q_y$ ) are consistently around 4 times higher for the ELM compared to the predicted Cramér-Rao lower bound.**

Configuration	$\sigma q_x$ (cm)		$\sigma q_y$ (cm)	
	Fisher	ELM	Fisher	ELM
Best CGA (Fisher)	2.11	9.13	2.11	9.69
Best K-means (Fisher)	2.01	9.08	1.97	8.62
Best Kohonen SOM (Fisher)	1.99	9.14	1.99	8.69
Best CGA (ELM)	2.18	8.50	2.19	8.17
Best K-means (ELM)	1.99	8.74	2.00	8.59
Best Kohonen SOM (ELM)	2.00	8.33	2.00	8.46
Fixed	1.98	8.70	1.98	8.65



**Figure 4.3: On the left the average  $\sigma$ s of the Fisher heatmaps are shown. On the right the  $\sigma$ s of the ELM error are shown. The figure shows that the ELM errors are larger than the Cramér-Rao lower bound.**

cording to the Fisher Fitness Function was used. Twenty different ELMs were trained and tested using the source dataset. The heatmap is generated from the prediction errors of the test sets. The errors are pooled into their closest grid point. This allows the standard deviations of each heatmap point to be calculated.

Figure 4.4 shows the Cramér-Rao heatmap and an ELM heatmap of this configuration. The heatmap shows the standard deviation of the position parameters  $q_x$  and  $q_y$ .

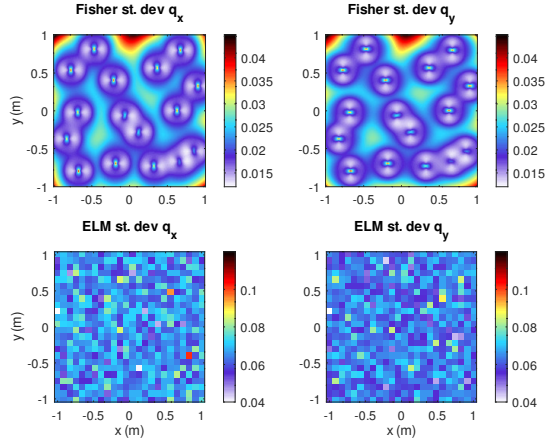
The heatmaps cannot be compared on the same scale because of the factor difference between Fisher and ELM (section 4.2). There appear to be some darker spots in the ELM but also some brighter spots. The dark corner at the top of the sensor array that is visible in the Fisher heatmap is not present in the ELM heatmap.

#### 4.4 Dark corners in sensor configurations

Three different metrics were used to compare the different sensor configurations (figure 4.5). For each sensor configuration, the Mean Squared Errors and Mean Euclidean Distance of the corresponding ELM were calculated. For the fixed configuration, 100 different ELMs were trained. Fisher Fitness scores for each sensor configuration were also calculated.

All the metrics seem to show that there are no big differences between the different configurations. This means that the amount and magnitude of the





**Figure 4.4: Fisher heatmaps compared to heatmaps that were generated from multiple sets of ELM errors. The scales of the colormaps are not the same because the ELM has larger errors! The dark corners visible in the Fisher heatmaps are not clearly visible in the ELM heatmaps.**

dark corners in these configurations should be approximately the same. Both the MSE and MED graphs show that there appears to be an outlying CGA configuration that is better than all other configurations for the ELM. It does not appear to be better according to the Fisher Fitness Function.

#### 4.4.1 Fisher Fitness Function

According to the Fisher Fitness Function scores (figure 4.6), the CGA configurations are decidedly worse than the other configurations. There is no overlap between the CGA scores and those of the other methods. The difference in scores between the K-means configurations, Kohonen SOM configurations and the grid configuration are small. There is a lot of overlap between them. There does not appear to be a configuration with a better score than the fixed/grid configuration.

#### 4.4.2 Mean Euclidean Distance

The Mean Euclidean Distance (figure 4.7) shows a larger overlap between the methods. The sensor noise does cause a variance in the performance of the ELMs. This is shown in the boxplot of the fixed configuration.

The CGA configurations, on average, seem to have higher errors than the other configurations. There is one outlying CGA configuration that has lower errors than all other configurations. There is almost no difference between the K-means and Kohonen SOM configurations as well as the fixed/grid configuration. Their distributions of the ELM Mean Euclidean Distance seems to be the same.

## 5 Conclusion

### 5.1 Fisher and ELM

The Extreme Learning Machine is capable of predicting the positions of moving sources via the sensor output of square artificial lateral line sensor configurations. The ELM errors are significantly larger than the Cramér-Rao lower bound. The dark corners that were visible in the Fisher information heatmaps do not appear in the ELM output.

### 5.2 Different configurations

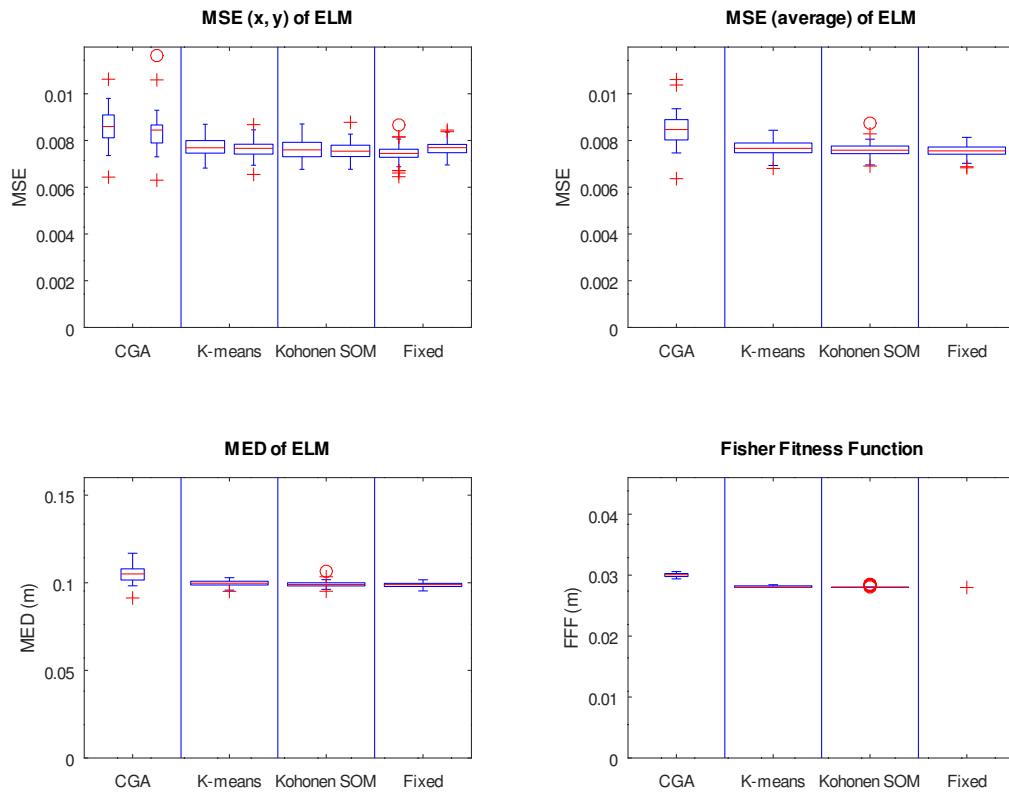
The fixed/grid configuration reduces the dark corners the most according to the Fisher Fitness Function. The CGA did not find the best configuration for this fitness function. The grid or grid-like sensor configurations reduced the dark corners the most when looked at with Fisher information heatmaps.

For the ELM the configurations that were generated by K-means and Kohonen SOM are approximately equal to each other. They are also comparable to the fixed/grid configuration. The best configuration was one of the CGA configurations.

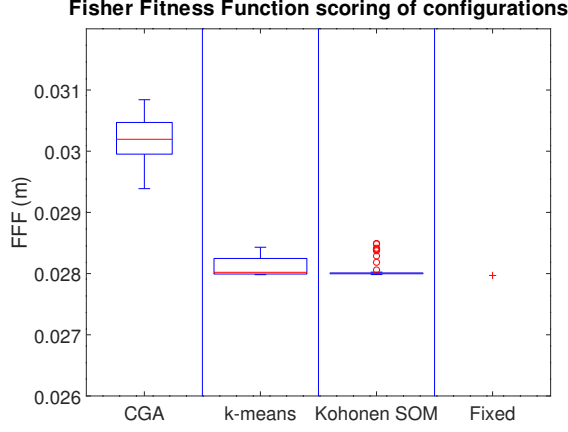
## 6 Discussion

### 6.1 Fisher and ELM

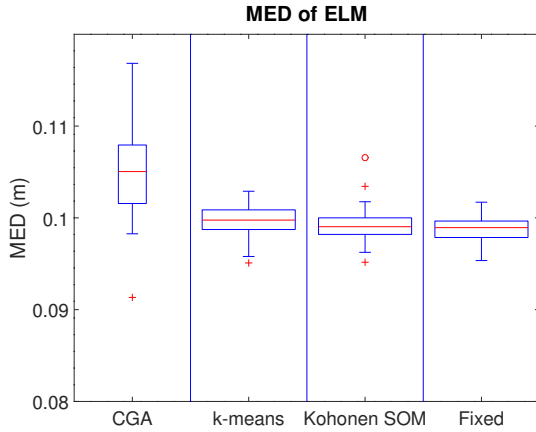
The result from the Fisher information matrix heatmaps seem to correspond to the previous study (Pirih and van Netten, 2018). In the previous study, there was also a dip in detection distance of the x and y components above the bi-axial sensor. This was in comparison to the volume above the sensor instead of a plane. The circumstances are not perfectly identical but the results seem in line with each other.



**Figure 4.5:** The different configuration scoring metrics for all configurations. All plots are scaled differently but all start at 0. They show the same trend, the CGA configurations are on average worse as they are associated with higher errors and scores.



**Figure 4.6: The Fisher Fitness Function scores of the different configurations from 4.5 but zoomed in. The scores of the CGA configurations are far appart from the others. No configuration clearly beats the fixed/grid configuration.**



**Figure 4.7: The Mean Euclidean Distance of ELM corresponding to the different configurations from 4.5 but zoomed in. There is heavy overlap between all the methods. The CGA configurations have, on average, higher MEDs. There is one CGA configuration that is significantly better than all other configurations (shown in figure 4.1, bottom left).**

The ELM shows no extreme dark corners compared to the Fisher heatmaps. This is in correspondence with earlier results. The error only increased when sources were incredibly close to the sensor line (Boulogne et al., 2017). In this experiment, sources cannot get closer than 25 centimeters to any sensor. Furthermore, there are always other lines that are still able to detect the source.

### 6.1.1 ELM dark corners

The heatmaps created from the ELM errors had poor resolution. This is because of the way that they were made. With increased resolution, actual dark corners could be found. This would not necessarily detect all dark corners. Dark corners that only occur for specific position and velocity vector pairs are hard to find with a random dataset of only 3000 samples.

The lack of dark corners may otherwise be attributed to the increased error of the ELM compared to the Cramér-Rao lower bound. The ELM may have produced a general solution that has fewer dark corners at the cost of some accuracy. A general solution would have the same or similar prediction errors for every case. There are no dark corners when the ELM prediction errors vary little over the source plane.

## 6.2 General performance of sensor configurations

In general, the grid sensor configuration and the grid-like sensor configurations that were generated by K-means and Kohonen SOM performed well. They had the lowest ELM errors and the best Fisher Fitness Function scores and so the least dark corners. The CGA, although it had the lowest scores and highest errors, was still within a single magnitude of the other methods.

It cannot be concluded that grid-like or evenly-spaced sensor configurations of artificial lateral line sensors are always the best option. No better configurations were found for the current setup. It would be interesting to look into other environments and whether or not the methods fare better or worse. It is easy to add the third dimension or velocities back into the Fisher matrices and other sensor spaces and source spaces can be used.

It is hard to say if this setup would also work for ships in ports. Potential flow does not model. The speeds and sizes of ships should, however, generate similar water flow patterns at larger distances. Whether this is the case and if ELM can also be used to detect model or real life ships is outside the scope of this research.

### 6.2.1 CGA

The implementation of the CGA deviates slightly from the source. Only the crossover and mutation functions were taken from this source. This may have reduced the effectiveness of the continuous genetic algorithm. It did supply a varied set of configurations for the other parts of the experiment.

The outlier configuration performed exceptionally well with the ELM but not with the Fisher Fitness Function. This sensor configuration and its sensor data may have fit the source dataset better than the other sensor configurations. As the same two sets of sources are used for the training and testing of every ELM, this can be a bias.

### 6.3 Kohonen SOM

The algorithm does not perform as in the source material due to the use of the actual distance between codes instead of the lattice distance. This occasionally caused codes and thus sensors to overlap. An enforced minimum on the distance could have prevented this. However the algorithm still gave many acceptable configurations.

## References

- 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, 2017.
- R. Chelouah and P. Siarry. A continuous genetic algorithm designed for the global optimization of multimodal functions. *Journal of Heuristics*, 6(2):191–213, Jun 2000.
- Branislava Ćurčić-Blake and Sietse M. van Netten. Source location encoding in the fish lateral line canal. *Journal of Experimental Biology*, 209(8):1548–1559, 2006. doi: 10.1242/jeb.02140.
- 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, Feb 1985.
- J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, Calif., 1967. University of California Press.
- 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, Dec 1980.
- Primož Pirih and Sietse M van Netten. Theory of hydrodynamic imaging: Potential flow in three dimensions. Part of a deliverable for the Lakshmi Project., 2018.
- Nicolas Rougier and Yann Boniface. Dynamic self-organising map. *Neurocomputing*, 74(11):1840–1847, May 2011 2011.
- Yingchen Yang, Nam Nguyen, Nannan Chen, Michael Lockwood, Craig Tucker, Huan Hu, Horst Bleckmann, Chang Liu, and Douglas L Jones. Artificial lateral line with biomimetic neuromasts to emulate fish sensing. *Bioinspiration & Biomimetics*, 5(1):016001, 2010.

## A Extra plots

Plots showing interesting information that was not covered in the paper. Figure A.1 is an outdated graph that shows how the sensor configuration methods placed the sensors. Figure A.2 shows what happens when the source plane is closer to the sensor plane so that the sensor to noise ratio is shifted toward the sensor. Figure A.3 shows what happens when the source plane is further away from the sensor plane so that the sensor to noise ratio is shifted toward the sensor noise.

## B Partial derivatives of water flow

Equations B.1 and B.2 show the partial derivatives of the Potential Flow model (from Pirih and van Netten (2018)). The direction  $m$  of the water flow indicates the axis over which the flow is measured. Similarly,  $p$  and  $n$  are the axes for the source position and source movement vector. Here 1 is the x-axis and 2 is the y-axis. The number 3 is reserved for the z-axis and can be used as well for both measuring and as component of the water flow or object movement.

$$\frac{\partial v_m}{\partial w_p} = -\frac{a^3}{2 * |r|^3} \left( \delta_{m,p} - 3 * \frac{r_m * r_p}{|r|^2} \right) \quad (\text{B.1})$$

$$\frac{\partial v_m}{\partial q_n} = -\frac{3 * a^3}{2 * |r|^4} \left( \frac{w_1 * r_n + w_n * r_1}{|r|} + \left( \delta_{1,n} - 5 \frac{r_1 * r_n}{|r|^2} \right) \left( \frac{w \cdot r}{|r|} \right) \right) \quad (\text{B.2})$$

$$\delta_{m,p} = \begin{cases} 1, & \text{if } m = p \\ 0, & \text{otherwise} \end{cases}$$

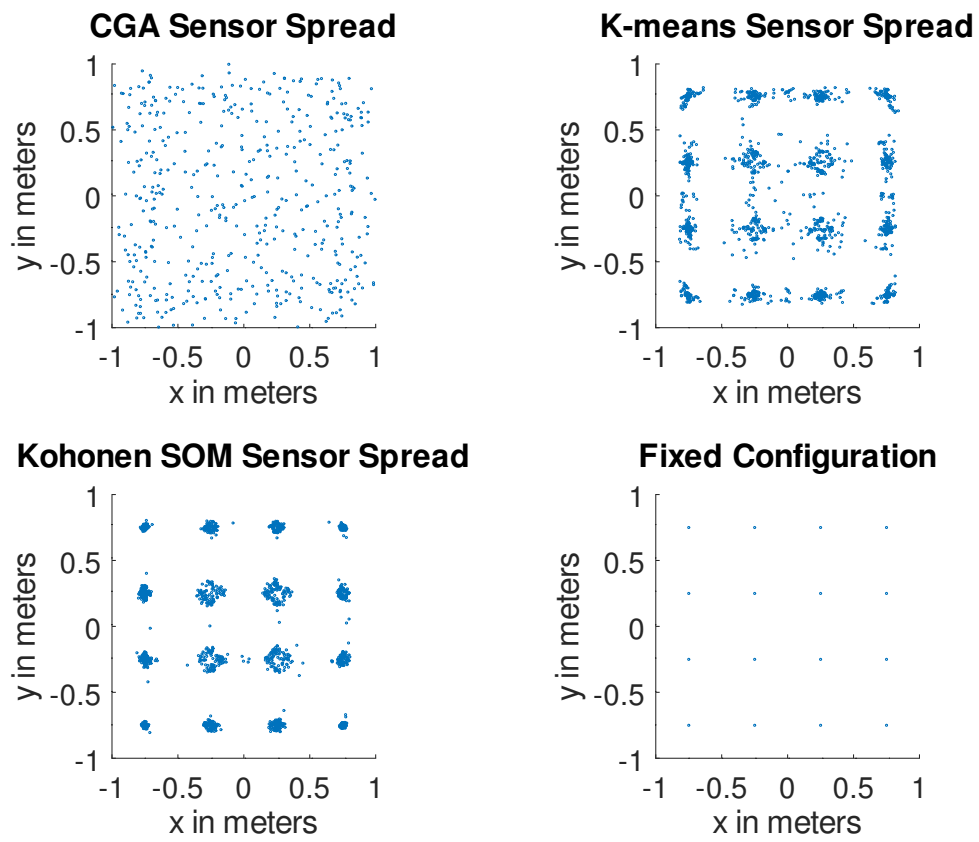
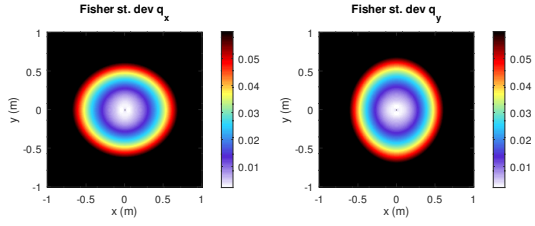
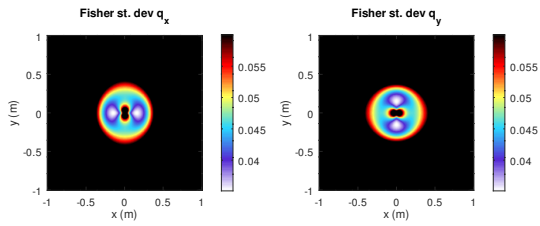


Figure A.1: The spread of the sensor positions that were generated by each algorithm.



**Figure A.2:** When the source plane is closer to the sensor plane the dark corner above the sensor disappears.



**Figure A.3:** When the source plane is further away from the sensor plane the dark corner above the sensor becomes larger.