Predicting How Fish Swim with Neural Networks

Cassidy Buhler
Department of Decision Sciences & MIS
Drexel University
Philadelphia, PA
Email: cb3452@drexel.edu

Shraman Kadapa
Department of Mechanical Engineering & Mechanics
Drexel University
Philadelphia, PA
Email: sk3496@drexel.edu

***Abstract*—In order to design and develop good fish robots, it is important to understand how fish uses its fin and body to produce thrust. Different fish species vary by their fin location, which can directly impact how much thrust it can produce. With different fin locations, the kinematics of the fin flapping can change. Fish can dynamically change their flapping frequency, and phase difference between fins based on different locomotive tasks. In this paper, we show how neural networks can be used to predict a fish's thrust and phase difference given the fish's flapping frequency, distance between fins, and lateral force. We find that our multi-output regression networks are unable to capture the complexity of the data, as phase difference is difficult to predict.**

I. Introduction

Fish use coordinated motion between its fins to swim and maneuver underwater. The location and kinematics of fish fins change based on different swimming tasks and fish species. The position and timing, known as the phase difference, flapping frequency and distance between fins affects the thrust production. The main objective of this work is to understand how fish uses its fins to propel itself. Specifically, we use neural networks with multi-output regression to predict a fish's phase difference and thrust, given the flapping frequency, distance between fins, and lateral force. In practice, a roboticist should be able to input the specifications of a fish and the model will tell them the maximum thrust that fish can provide, and the phase difference at which this thrust is attained.

II. Related Work

The overarching goal of this work is to have good control of underwater fish robots so that they can swim and maneuver well for ocean exploration, unlike other contemporary unmanned underwater vehicles (UUV). With good control in propulsive forces, the robots can be used efficiently for various exploratory tasks such as pollution tracking, mining, and search and rescue to name a few. Previous studies have shown that tuning the phase difference between the fins can greatly affect the propulsive forces produced by finned robots [9]. This is caused due to changes to the fin-wake interaction between fins. As the distance between fins and the flapping frequency are changed, the wake shed by the upstream fin as it interacts with the downstream fin changes. Even though there have been many investigations of forces produced by single and multiple fins from bio-robots, the effect of location and kinematics of fish fins on propulsive forces and the associated wake produced is not well understood [6]–[8], [12]. Due to its uncertain environment, it gets difficult to design and model such underwater robots. Most of the current work in this domain predominantly relies on designing controllers or modelling the system dynamics using deep learning and reinforcement learning [10], [13], [14]. There are very few studies that exploit the fish swimming gait using neural networks. With unknown nonlinear dynamics in the system, it is important to use neural networks to understand different maneuvers that are achievable by fish robots [2]. This work will help roboticists design, develop and control bio-robots that can swim effectively.

III. Data

A. Data Collection

A computational fluid dynamic simulation COMSOL [1] has been developed to study the interaction between two fish fins. The data is simulated by varying a fish's phase difference between 0-360 degrees, flapping frequency and horizontal distance between the fins. At each phase difference selected, we record the thurst and lateral forces produced by the fish fins for 5 fin cycles. Since the thrust can vary over every fish fin beat, we record the average thrust from these 5 fin cycles. Each fish is tested for various phase difference values, and the phase difference that yields the maximum average thrust is denoted as the optimal phase difference.

We obtained the 3 features (frequency, distance, force) for 460 fish which yields a 460×3 feature matrix X. Since we predict two features–phase difference and thrust–the target output is a matrix y of size is 460×2. A correlation matrix and pairwise comparisons of the data are included in Figure 1 and Figure 2, respectively.

B. Preparing Data

We partitioned the data into training, validation, and testing sets using a 80-10-10 split. This gave 368 fish for training, 42 fish for validation, and 42 fish for testing.

Once the data was divided, we standardized the features to be within the range $(-1, 1)$ using MinMaxScaler from scikit-learn [11] based on the maximum and minimum values from the training data. In addition, the target data was also standardized from its training set. Given that the data is in various units, normalizing the data was necessary to avoid numerical errors in the model. However, once we train and

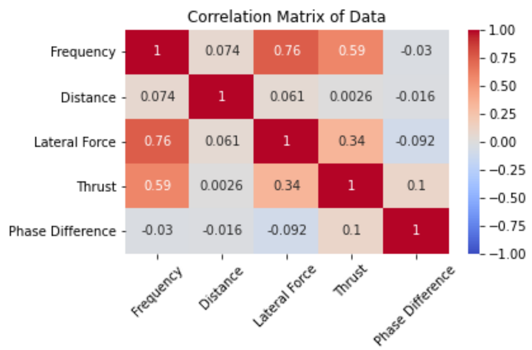

Fig. 1. Correlation matrix of the datatset.

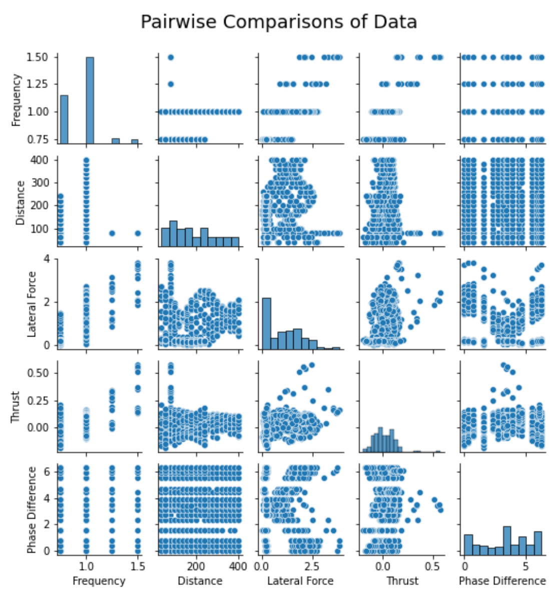

Fig. 2. Pairwise plots of each feature in the dataset. Frequency is measured by hertz. Distance denotes the distance between fins. Thrust and lateral forces are in newtons. Phase difference is in radians.

make predictions on the model, the predictions and testing data is reverted back to original scale when evaluating model accuracy. Although the standard scaling method is converting data to a standard normal distribution, we chose to use MinMax scaling because some of our features didn't lie on a normal distribution and are on a bounded interval.

IV. Methodology

Designing a neural network can be complex, as there are many different modifications and hyperparameters to tune. For this reason, we created 3 different neural networks to test on our data. NN1 was created first, then we developed NN2 based on weaknesses we saw in NN1. The third network NN3 was developed to find a middle ground between NN1 and NN2.

In addition, we used Keras [3] to create the neural networks. For each network, we use the Adam [5] optimizer. The shared parameters of our networks are outlined in Table I.

Parameter	Value
Max Epoch	500
Batch Size	16
Optimizer	Adam
Activation Function	ELU
Loss Function	MSE

TABLE I
Parameters used for all three networks.

A. Neural Network 1 (NN1)

The first network contains 5 layers: input layer, 3 hidden layers, and the output layer. The number of nodes in each layer are 3, 20, 20, 10, and 2 respectively. Every layer uses the exponential linear unit (ELU) activation function, except the output layer, which uses the linear activation function. The learning rate for this network was 0.0001.

Parameter	Value
Layers	[3,20,20,10,2]
Learning Rate	0.0001

TABLE II
NN1 architecture and parameters details.

B. Neural Network 2 (NN2)

The next network includes more nodes than the previous, in order to help the model learn better. However, to prevent overfitting, we included an alpha dropout layer, which drops data based on a given probability, while preserving the mean and standard deviation so the dropout doesn't impact the scaling of the data. The probability we used was 0.25.

Parameter	Value
Layers	[3,50,Dropout,40,30,2]
Learning Rate	0.0001

TABLE III
NN2 architecture and parameters details. Dropout layer was with probability 0.25.

C. Neural Network 3 (NN3)

The last network was created to find a middle ground between NN1 and NN2. We used more nodes than NN1, but less than NN2. The learning rate was also decreased based on the results from NN2.

Parameter	Value
Layers	[3,40,20,10,2]
Learning Rate	0.000025

TABLE IV
NN3 architecture and parameters details.

V. Results

As displayed in Figure 3, each network did well at minimizing the loss function for the training data. We see that the validation set minimized best in NN2, however the training set in this network seemed to oscillate which would imply the learning rate is too high [4]. For this reason, we decreased the learning rate in the NN3 to avoid this issue.

The network output will be evaluated based on the mean squared error (MSE). For $i = 1, \dots, n$ where n is the total

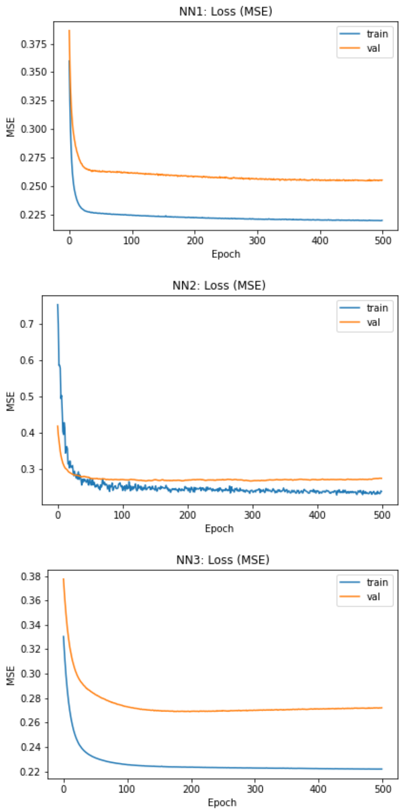

Fig. 3. Loss functions during training for each network.

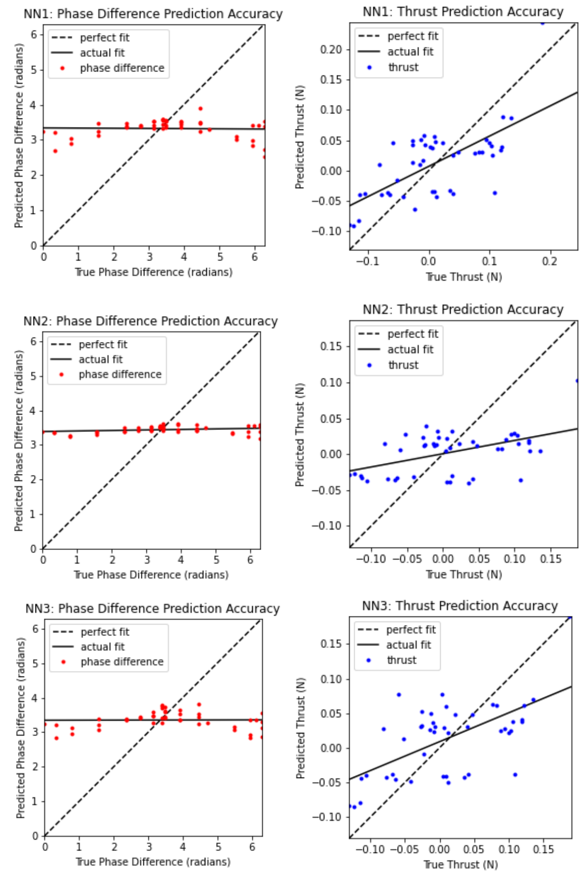

Fig. 4. Each row is the prediction results for the 3 neural networks.

number of fish and $j = 1, \ldots, m$ where m is the number of outputs, we define y_{ij} and $\hat{y}_{ij}$ as the true and predicted values, respectively. Note that in our case where we have 2 values to predict, thrust and phase difference, therefore $m = 2$. MSE is defined as the following:

$$\text{Multi-Output MSE} = \frac{1}{n * m} \sum_{i=1}^{n} \sum_{j=1}^{m} (y_{ij} - \hat{y_{ij}})^2 \quad (1)$$

To compute the MSE of a single feature's predictions, we use the modified formula of the above.

$$\text{Single-Output MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y_i})^2 \quad (2)$$

We measured the MSE of the predictions given from all three models. Given that we had two outputs, we computed the overall MSE as well as the individual MSE of each feature. These are presented in Table V.

The first network predicted thrust the best, but had the worst phase difference out of the networks. The second network improved on phase difference but had a drop in thrust MSE. The last network was a compromise between the first two, and had similar thrust MSE to NN1 while reduced phase difference MSE to NN2. All three networks had difficulties with predicting phase difference, and with the MSE of around 3 radians.

Model	Overall MSE	Thrust MSE	Phase Difference MSE
NN1	1.5285	0.0033	3.0537
NN2	1.4167	0.0047	2.8288
NN3	1.4952	0.0038	2.9866

TABLE V

MEAN SQUARED ERROR FOR EACH MODEL. OVERALL MSE, IS COMPUTED WITH BOTH THRUST AND PHASE DIFFERENCE PREDICTIONS BY EQUATION 1. THE LATTER TWO COLUMNS ARE GIVEN BY EQUATION 2.

In addition, we included prediction plots of the three networks in Figure 4. On the x-axis, we plotted the true value and the y-axis was the predicted value. A perfect prediction would have the solid black line match the dotted black line. We see that the thrust predictions are close to the dotted black line, but the phase difference predictions lack variation and hardly match the true value.

VI. Conclusion

The idea behind this project was to use neural networks to understand the swimming gait which is so efficiently used by various species of fish. There are many mechanical and physical properties that change based on swimming gait and various fish species respectively. Some of these factors are the distance between the fins, the phase difference between fins and flapping frequency of the fins. Based on our results, we found that phase difference was the most difficult to predict.

In addition, the thrust feature is only relevant when coupled with the phase difference, so an incorrect phase difference may yield the thrust prediction invalid. We hope to improve upon this in future work. The data collected for this study was not uniformly distributed across all parameters tested. Specifically, there were more data points for 0.75Hz and 1 Hz frequency as compared to other frequencies. With a more wider range of resolution in data, we aim to accurately predict thrust produced and the phase at which fins must be flapped to produce maximum thrust. The use of neural networks can greatly influence this research since there are many parameters involved in fish swimming and these factors are coupled which makes it harder to understand using experiments and simulations. Once we have enough uniformly distributed data points across all parameters, we aim to optimize for lateral force in order to swim rectlinearly.

VII. BIBLIOGRAPHY

REFERENCES

[1] COMSOL Multiphysics® v. 5.6. www.comsol.com. COMSOL AB, Stockholm, Sweden.

[2] R. Amin, A. A. Khayyat, and K. G. Osgouie. Neural networks modeling of autonomous underwater vehicle. In *Proceedings of 2010 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, pages 14–19, 2010.

[3] François Chollet et al. Keras. https://keras.io, 2015.

[4] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[6] George V Lauder. Swimming hydrodynamics: ten questions and the technical approaches needed to resolve them. In *Animal Locomotion*, pages 3–15. Springer, 2010.

[7] George V Lauder. Fish locomotion: recent advances and new directions. *Annual review of marine science*, 7:521–545, 2015.

[8] George V Lauder and Peter GA Madden. Learning from fish: kinematics and experimental hydrodynamics for roboticists. *International journal of automation and computing*, 3(4):325–335, 2006.

[9] Anthony P. Mignano, Shraman Kadapa, James L. Tangorra, and George V. Lauder. Passing the wake: Using multiple fins to shape forces for swimming. *Biomimetics*, 4(1), 2019.

[10] Jianjun Ni, Liuying Wu, Pengfei Shi, and Simon X. Yang. A dynamic bioinspired neural network based real-time path planning method for autonomous underwater vehicles. *Computational Intelligence and Neuroscience*, 2017:9269742, Feb 2017.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[12] R.E. Shadwick, A.P. Farrell, and C.J. Brauner. *Physiology of Elasmobranch Fishes: Structure and Interaction with Environment*. ISSN. Elsevier Science, 2015.

[13] Jiemei Zhao. Neural network predictive control for autonomous underwater vehicle with input delay. *Journal of Control Science and Engineering*, 2018:2316957, May 2018.

[14] Yi Zhu, Fang-Bao Tian, John Young, James C. Liao, and Joseph C. S. Lai. A numerical study of fish adaption behaviors in complex environments with a deep reinforcement learning and immersed boundary–lattice boltzmann method. *Scientific Reports*, 11(1):1691, Jan 2021.