Embodied hydrodynamic sensing and estimation using Koopman modes in an underwater environment

Colin Rodwell Phanindra Tallapragada

Abstract— **Objects moving in water or stationary objects in streams create a vortex wake. An underwater robot encountering the wake created by another body experiences disturbance forces and moments. These disturbances can be associated with the disturbance velocity field and the bodies creating them. Essentially the vortex wakes encode information about the objects and the flow conditions. Underwater robots that often function with constrained sensing capabilities can benefit from extracting this information from vortex wakes. Many species of fish do exactly this, by sensing flow features using their lateral lines as part of their multimodal sensing. Besides the necessary sensing hardware, a more important aspect of sensing is related to the algorithms needed to extract the relevant information about the flow. This paper advances a framework for such an algorithm using the setting of a pitching hydrofoil in the wake of a thin plate (obstacle). Using time series pressure measurements on the surface of the hydrofoil and the angular velocity of the hydrofoil, a Koopman operator is constructed that propagates the time series. The Koopman modes associated with this operator are then used by a convolutional neural network (CNN) to estimate the distance and location of the plate. The hydrofoil (or a bioinspired robot) thus acquires the capability to 'blindly' sense obstacles using time-varying pressure measurements.**

I. Introduction

The locomotion of fish and other aquatic swimmers has many desirable characteristics such as energy efficiency, agility, and stealth [1], [2], which have inspired mimicry in bioinspired robots [3], [4]. Closely related to and aiding the locomotion is the ability of fish to sense and process the spatiotemporal information in the water around them. Objects moving in water or stationary objects in streams create a vortex wake. An underwater robot encountering the wake created by another body experiences disturbance forces and moments. These disturbances can be associated with the disturbance velocity field and the bodies creating them. Essentially, information about fluid flow and the objects that create these flows is encoded in the spatiotemporal evolution of the vortical structures, whether the bodies creating them are cylinders, hydrofoils, underwater robots or fish. Underwater robots that often function with constrained sensing capabilities can benefit from extracting this information from vortex wakes. Many species of fish do exactly this, by sensing flow features using their lateral lines as part of their multimodal sensing [1], [5], [6].

This work was supported by grant 2021612 from the National Science Foundation and grant 13204704 from the Office of Naval Research.

Colin Rodwell and Phanindra Tallapragada are with the Department of Mechanical Engineering, Clemson University, Clemson, SC, 29631, USA. {crodwel@g.clemson.edu, ptallap@clemson.edu}.

The complexity and high (infinite) dimensionality of fluid flows around a swimmer present significant challenges to emulate fish-like hydrodynamic sensing and extract the relevant information from sensor data of the flow. This particular challenge is not restricted to bioinspired fish-like swimmers, but has been present in the broad areas of fluid flow estimation, model reduction, and active flow control. Proper orthogonal decomposition (POD) [7], [8] and the gappy POD [9] have been tools for model reduction in turbulent flows for decades, and have also been applied for unsteady flow sensing past an hydrofoil and estimation of surface pressure [10], [11]. Model reduction of complex flows using the Koopman operator approach has extended the POD approach to a dynamical systems framework [12], [13]. Subsequent developments in the application of machine learning in dynamical systems have created algorithms for learning the dynamic modes or Koopman modes of a dynamical system from often sparse data [14]–[16]. Similar methods combining machine learning with dynamical systems are increasingly playing an important role in model reduction in fluid mechanics [17]–[24]. Flow estimation in the near field of a body that combines traditional filtering approaches with the lifted dynamics induced by the associated Koopman operator have been studied recently in [25], [26].

This paper considers a different but related problem motivated by underwater robots where on-board sensors such as IMUs and pressure sensors can usually measure only dynamic and kinematic variables of the robot itself and not measure the ambient velocity field. We consider the problem of the estimation of the spatial location of an up stream obstacle in a flow past a pitching hydrofoil. It is assumed that pressure on the surface of the hydrofoil can be measured at fixed locations on the body along with the pitch angular velocity of the hydrofoil. Using time series pressure measurements on the surface of the hydrofoil and the angular velocity of the hydrofoil, a Koopman operator is constructed that propagates the snapshots of data forward in time. The modes from a spectral decomposition of this operator are then used to create 'images'. These images are input to a convolutional neural network (CNN) to estimate the distance and location of the plate (obstacle). The hydrofoil (or a bioinspired robot) thus acquires the capability to blindly sense obstacles using proprioception. Segments of an underwater could then act as sensors and flow controllers [27].

II. Embodied sensing and estimation of obstacles

The simulation setup for the estimation problem is shown in Fig. 1. A fish shaped body (a NACA 0018 hydrofoil) is

pinned at a distance b downstream from a thin flat plate. The hydrofoil's pitching motion is restrained by a torsional spring of stiffness k and a dashpot of damping coefficient c. A free stream velocity U_∞ is prescribed to as shown. Pressure is measured in the simulation at N points in the surface of the hydrofoil; these 'pressure sensors' are distributed uniformly around the length of the hydrofoil. Furthermore the angular velocity $\Omega(t)$ of the hydrofoil is also measured. The hydrofoil pitches in response to both the free stream velocity as well as the disturbance wake created by the plate. The effects of disturbance wake produced by the flat plate obstacle are felt in the pressure and angular velocity measurements. The estimation problem considered in this paper is to estimate the values of b and d.

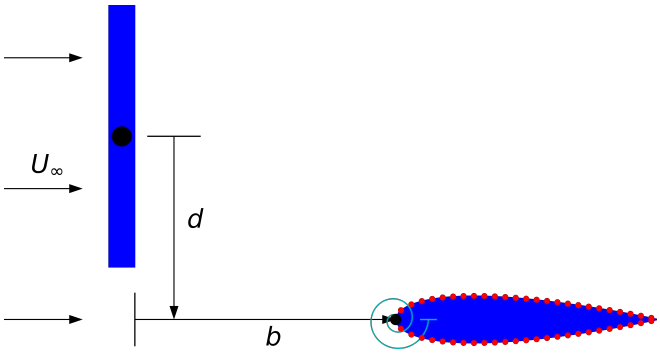


Fig. 1: A hydrofoil of length 1 m is pinned at its leading edge a distance of b downstream from an obstacle of height 1 m and thickness 0.1 m. The upstream obstacle is placed at a height of d from the center line. A torsional spring of stiffness $k = 1\ Nm/rad$ and damping $c = 1\ Nms/rad$ resists rotation of the foil. Pressures are recorded at the 50 points indicated on the surface of the foil.

A. *Fluid flow simulations and data generation*

The coupled dynamics of the fluid flowing past the plate and the hydrofoil are simulated using *OpenFOAM*®. The hydrofoil (NACA 0018) has chord length of 1 m and thickness of 0.18 m; the flat plate is also of length 1 m and is 0.1 m wide. A two-dimensional computational domain of length 6.2 m and height 4.4 m is constructed with a foil pinned at its leading edge to a point in the middle of the domain at a distance of 3.2 m from the left edge of the domain. The top and bottom of the domain have a slip boundary, but allow no normal velocity; the left edge (where the fluid enters the domain) is chosen to have a constant pressure of $p = 0$ and $U_\infty = 20$ m/s. The foil and obstacle both have no-slip and no-normal velocity boundary conditions. These conditions serve to simulate a portion of a water tunnel environment. The fluid is chosen to have unit density and Newtonian shear with kinematic viscosity $\nu = 1 \times 10^{-6} m^2/s$.

The simulation of this environment takes place in several steps. Before the foil and plate geometry are introduced, a rigid sparse square mesh with edge length 0.043 m is generated across the empty computational domain with the *blockMesh* utility. From there the bodies are introduced and a finer boundary mesh with edge lengths near the body of roughly 0.0056 m is generated and 'snapped' to them with the *snappyHexMesh* utility. To accurately capture the boundary layer, the local mesh is divided into 3 layers which become increasingly sparse at a distance from the body, until they are spliced into the block mesh at the outer boundary, resulting in a total of roughly 400,000 mesh points. A steady-state flow field is then approximated by fixing the foil rigidly and applying 3000 iterations of the SIMPLE (Semi-Implicit Method for Pressure Linked Equations) algorithm to the mesh using *simpleFoam*. This flow field is taken as an initial condition for the transient solution with foil motion solved using the *pimpleFoam* implementation of the PISO (Pressure-Implicit with Splitting of Operators) algorithm. Mesh motion is calculated automatically, with meshes within range 0.3 m of the body translating rigidly with the body, meshes at greater than 1 m distance remaining stationary, and meshes in the 0.3 m to 1 m region deforming without topology change. Three seconds of simulation data are generated and stored at intervals of 0.01 seconds. The data include the foil angular velocity and 50 pressure readings interpolated to fixed points on the foil surface (indicated in Fig. 1). The simulations are performed on a parametric grid where b varies uniformly from 0 m to 1 m and d varies uniformly from 1 to 3m, resulting in 121 simulations for the 121 pairs of (b, d). Additional test data is then generated at 50 random pairs $(b, d) \in [0, 1] \times [2, 3]$.

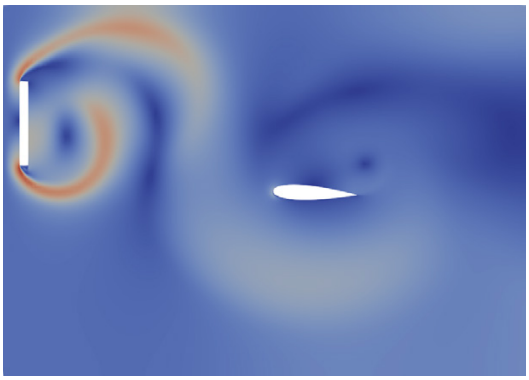
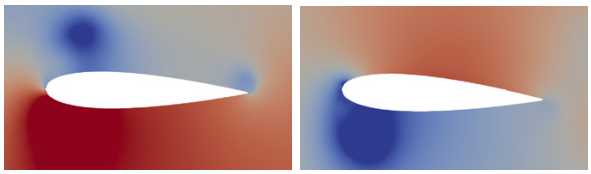


Fig. 2: The magnitude of the velocity field with $b = 3$ m and $d = 0.8$ m at time 0.43 s. Periodic vortex shedding from the plate is evident, and the interaction between that wake and the foil leads to secondary vortices being shed off of the foil.

B. *Koopman modes*

The motion of the fluid has complex patterns with some clear periodic features in a visual inspection of the pressure field as shown in Fig. 3, as well as some patterns that are difficult to discern. The underlying dynamics of the coupled foil-fluid are very high dimensional. The objective is to take a very small set of observables, specifically surface pressure and angular velocity of the foil, in the estimation of the parameters b and d. Knowledge of the



(a) (b)

Fig. 3: The static pressure profile around the foil with $b = 3$ m and $d = 0.8$ m at (a) t=0.17 s and (b) t=0.44 s. Interaction with the upstream wake causes substantial variation in the local pressure field, both spatially and temporally.

velocity field of the fluid is not assumed. Model reduction and reconstruction of reduced order models in such complex dynamical systems where only limited data on observables is available can be possible via the framework of the Koopman operator, a topic that has attracted much attention in recent years [14]–[16], especially in the context of fluid dynamical systems [28], [29]. Denoting the observables as $g(t) = [P_1(t), ..., P_N(t), \Omega(t)]^T$ where $P_i(t)$ denotes the pressure on the foil at location i at time t and $\Omega(t)$ denotes the foil angular velocity similarly at time t, the Koopman operator $\mathcal{K}$ is a linear operator $\mathcal{K} : L^2 \mapsto L^2$ and propagates the observables forward in time

$$\mathcal{K}_T g(t) = g(t + T). \tag{1}$$

Because in experiments and simulations the observables are measured at discrete times $(t_0, t_1, ..., t_s)$, we denote the observables as $[g(t_1), ..., g(t_s)]$ where a projected Koopman operator $\mathcal{K} : \mathbb{R}^{N+1} \mapsto \mathbb{R}^{N+1}$ can be redefined as one that propagates the observables forward in discrete time

$$\mathcal{K} g(t_m) = g(t_{m+1}). \tag{2}$$

Due to linearity, $\mathcal{K}^m g(t_0) = g(t_m)$. The definition of the projected Koopman operator is not unique and is only one of convenience, as the projected Koopman operator can also be defined to act on any other finite dimensional subspace of L^2. The projected operator $\mathcal{K}$ cannot usually propagate the observables without any error, and a choice of an approximate operator is made such that $||\mathcal{K} g(t_m) - g(t_{m+1})||$ is small, under a norm $||\cdot||$.

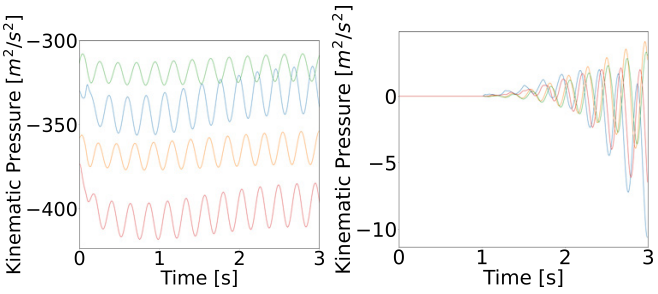
In the time series data generated by the OpenFOAM simulations, the observables are measured at equally spaced time intervals, $\Delta t = t_{m+1} - t_m = 0.01$ s for all $0 \leq m < s$. Assembling a matrix of m sequential time snapshots of the observables as $G(i) = [g(t_i), ...g(t_{i+m})]$ for any $i \geq 0$ and $m < s - i$, the Koopman operator propagates these sequential snapshots of observations as

$$G(i + 1) = \mathcal{K} G(i). \tag{3}$$

The operator $\mathcal{K}$ can then be redefined as one that minimizes the matrix norm $||\mathcal{K} G(i) - G(i + 1)||$. We can then find $\mathcal{K}$ that best fits the data as

$$\mathcal{K} = G(i + 1) \;\; G(i)^+ \tag{4}$$

where A^+ indicates the Moore-Penrose pseudoinverse of A. Figure 4(a) shows the time series data for pressure at 4 arbitrary locations and Fig. 4(b) shows the error in the propagated values of these pressure observations where $\mathcal{K}$ is constructed with $i = 0$ and $m = 99$, i.e., $G(0) = [g(t_0), ..., g(t_{99})]$ and $G_1 = [g(t_1), ..., g(t_{100})]$. The error in the propagated values of the pressure is within 3% even for future times as large $k = 200$ instants (2 seconds).



(a) (b)

Fig. 4: (a) A sampling of rows of G corresponding to pressure data at select sensors for b=2.7 m and d=0.8 m and (b) the error in an approximation of those pressures constructed by from the pressure at $t = 0$, p_0, as $\mathcal{K}^n p_0$ for timestep n. The Koopman operator is constructed using only the first second of pressure values, but continues to fit the data well beyond that point, indicating capture of the underlying dynamics beyond simple curve fitting.

The spectral decomposition of the Koopman operator $\mathcal{K}$ is used in the subsequent machine learning. This spectral content is obtained via the singular value decomposition

$$U \Sigma V^* = \mathcal{K} \tag{5}$$

with the diagonal elements of $\Sigma_{ii} > \Sigma_{jj}$ if $i > j$. The columns of matrices U and V then form an orthogonal basis for $\mathbb{R}^{N+1}$ and $\mathbb{R}^m$ respectively. We choose $m = 100$, i.e. 1 second of measured sensor data, to estimate $\mathcal{K}$. Associated with a $G(i)$, an operator $\mathcal{K}(i)$ and its associated singular vectors $U(i)$ and $V(i)^*$ and singular values Σ^i are computed. Figure 5 shows the image $U(1)$ from one of the simulations. A clear pattern of horizontal bands is evident, the shape of which is unique to these flow parameters. Such features in the spectral content of the Koopman operator can be extracted and identified using convolutional neural networks, a standard method in identifying features in images.

C. Convolutional neural network

Convolutional neural networks (CNNs) are an algorithm modeled after biological vision processing [30] that has seen recent success in problems with high-dimensional input such as image and sound classification. They operate based on kernels, which are dense neural networks (DNNs) that can only see data in a limited window. Those windows are then moved along the input and the kernel output for each window becomes the input to the next layer. The relatively low number of parameters involved compared to DNNs allows for

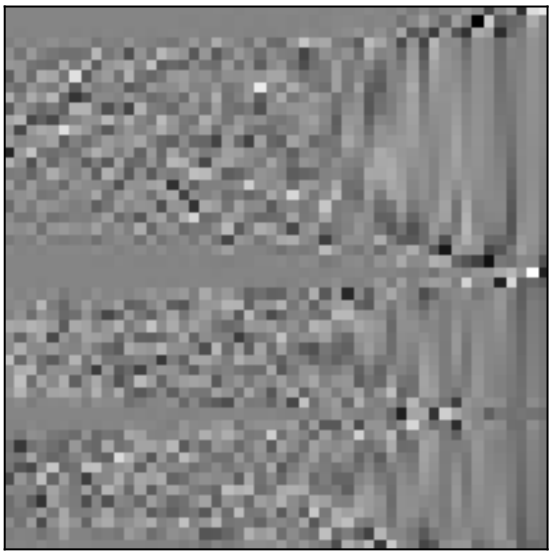


Fig. 5: The left singular matrix $U(1)$ for $b = 2.739$ m and $d = 0.481$ m. A clear pattern of horizontal bands is evident, the shape of which is unique to these flow parameters.

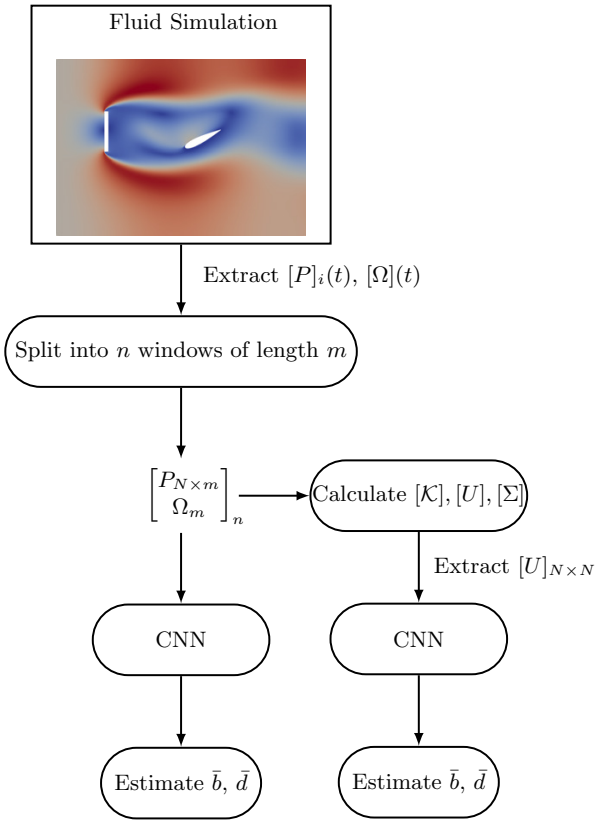


Fig. 6: The overall classification algorithm. A fluid simulation is performed and the pressure and velocity data extracted. This data is then combined into a set H^k, which are the inputs to a CNN classifier. The set H^k is also processed into the Koopman spectral input E^k, which are similarly passed as inputs to a similar CNN classifier.

better generalization of the data, and reducing the operation to multiple windows offers equivariance to shifted input. This property is particularly useful in processing periodic time-series data such as the pressure data here: The initial phase of the pressure for any given simulation is arbitrary, and a slight difference in that phase can cause a large difference in the magnitude of pressure at a given point in time. DNNs, which work primarily by analyzing the magnitude of an input, are challenged by this, but CNNs, which focus on the relationship between near points, can perform well.

From each simulation with corresponding parameter values (b_k, d_k) we choose 40 unique time windows of data that are 1 second long (100 consecutive time snapshots) and construct the matrices $G^j(i)$ for $j = 1, ..., 40$ and where i is randomly chosen. Because a value of j identifies a specific time instant t_i, is dropped in further references to $G(i)$. The set of inputs for the CNN from the kth simulation is $H_k = (G_k^1, ..., G_k^{40})$ with parameters (b_k, d_k). The shift in input phase between the 40 windows helps to train the CNNs to identify inputs of unknown phase and reduces the risk of overfitting. The set of inputs H^k are used to directly train CNNs to establish a baseline case for estimation of the obstacle's location using direct sensor measurements without using the spectral content of a Koopman operator. For each set of inputs $H^k = (G_k^1, ..., G_k^{40})$ the corresponding set of left singular vectors of the Koopman operator $E^k = (U_k^1, ..., U_k^{40})$ is calculated. The set of inputs E^k are used to train another CNN to estimate the pair (b_k, d_k). The overall framework for the estimation of the parameters (b_k, d_k) is shown in Fig. 6. The left and right parallel estimates of (b_k, d_k) in Fig. 6 use direct sensor measurements H^k versus the spectral content E^k of a Koopman operator respectively, allowing for a comparison between the two approaches.

Hyperparameters for the CNNs are selected through a trial-and-error approach, and the same parameters are used on all the varieties of images. The first layer of both features 5 kernels with Rectified Linear Unit (ReLU) activation functions that operate on a 7×7 input window, with no padding at the edges, generating 5 convolved images. A pooling operation is then performed on each image, splitting each into 2×2 grids and keeping only the largest value from each. This convolution-pooling routine is then repeated again with the same parameters before the results are flattened into a 1-dimensional tensor. That tensor is then processed by a 4-layer dense neural network, with ReLU activation and 200, 100, 100, and 50 nodes, respectively. The estimates for b and d are then chosen as a linear superposition of the outputs of the final dense layer. The parameters of this network are optimized to minimize the mean square error of the estimation on the training set, which corresponds to 40 images per each of 121 simulations for a total of 4,840 images per network. The accuracy of the network is then evaluated on 50 designated test simulations again with 40 images each, and the saved estimate for each simulation is the average estimate over its images.

III. Results

Figure 7 shows the results of the estimation on the test data set. The CNN trained on the matrices $E^k = (U_k^1, ..., U_k^{40})$ that contain the spectral content of the Koopman operator vastly outperforms the CNN trained using direct sensor measurements. In general, the parameter d is easier to estimate than b, which may be intuitive considering the underlying dynamics: wakes tend to maintain their shape as they travel downstream which confounds the estimation of b, but the pressure profile changes greatly normal to the direction of wake translation, which should make the data more sensitive to changes in d and thus make the estimation simpler.

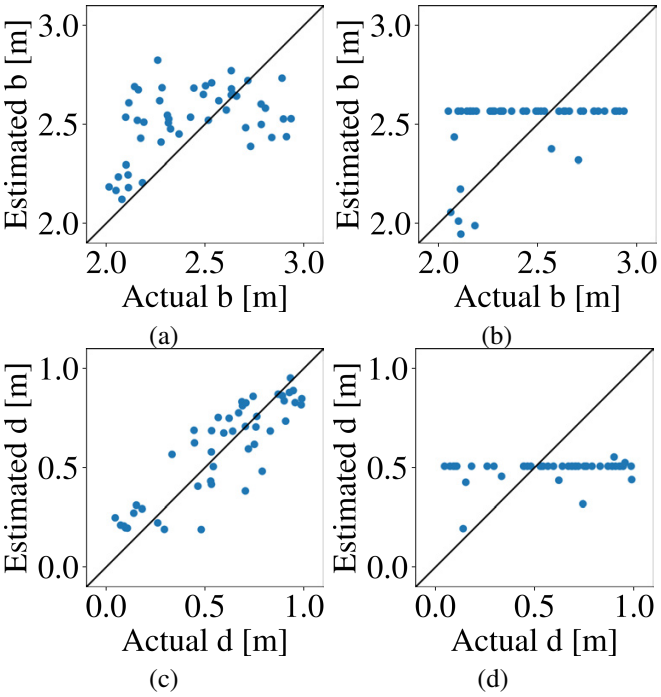


Fig. 7: Estimated vs. true values using (a,c) the set of E matrices and (b,d) the set of H matrices.

The estimation of the parameter b has comparatively large errors in the range where $2.25 < b < 2.5$. Incorporating angular velocity Ω into the homogeneous data input G creates difficulties for the CNN since the angular velocity is at least an order of magnitude smaller than the pressure. As a downside of their generality, CNNs lose this ability to treat parts of the data differently: because the same kernel is used for every window, Ω must be processed with the same weights as a pressure, even though it has different units, scale, and physical meaning. This corrupts the output of the lowest row of windows, which propagates through the multiple CNN and pool layers to corrupt the output. More importantly, the angular velocity of the pitching foil is merely the time integral of the angular acceleration of the body which in turn is the result of surface integral of the hydrodynamic moment due to the pressure on infinitesimal lengths of the surface of the body. This provides no new information and perhaps only confuses a CNN by incorporating a scaled spatiotemporal mean of the pressure measurements as an independent measurement.

To check if the estimation accuracy improves with the exclusion of the angular velocity input, two alternative CNNs were trained. The inputs for these CNNs were in one case direct pressure measurements, i.e. H^k but without the last row of measurements of the angular velocity. In the second case the input was the set singular vectors $F^k = (u_k^1, ..., u_k^{40})$, where the singular vectors u are computed from the SVD of the Koopman operator that propagates only pressure observables. The results of the estimation of the parameters (b, d) in test data using these CNNs is shown in Fig. 8. Once again the CNN trained on inputs based on the singular vectors of a Koopman operator vastly outperforms the CNN trained on inputs that are direct sensor measurements. More interestingly, excluding the angular velocity from the observables improves the estimation accuracy. This improved accuracy is seen in slightly better estimation in the values of d which lie closer to the straight line of unit slope and improves the estimation of b in the range $(2.5, 3)$.

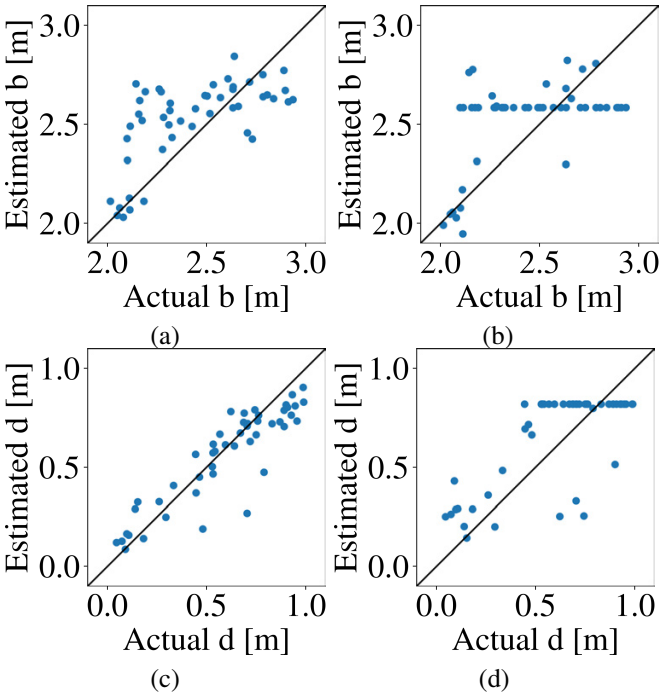


Fig. 8: Estimated vs. true values using (a,c) the U matrix and (b,d) the G matrix, both constructed purely with pressure data. The Koopman operator approach vastly outperforms direct estimation of the parameters (b, d) from sensor measurements.

The superior estimation ability of the Koopman operator approach remains even when noise is added to the sensor measurements as shown in Fig 9. Each pressure and angular velocity measurement is perturbed by a 2% noise sampled from Gaussian distribution with zero mean and unit variance. The Koopman operator essentially filters the noise and reproduces estimates that are close to the case where the measurements have no noise.

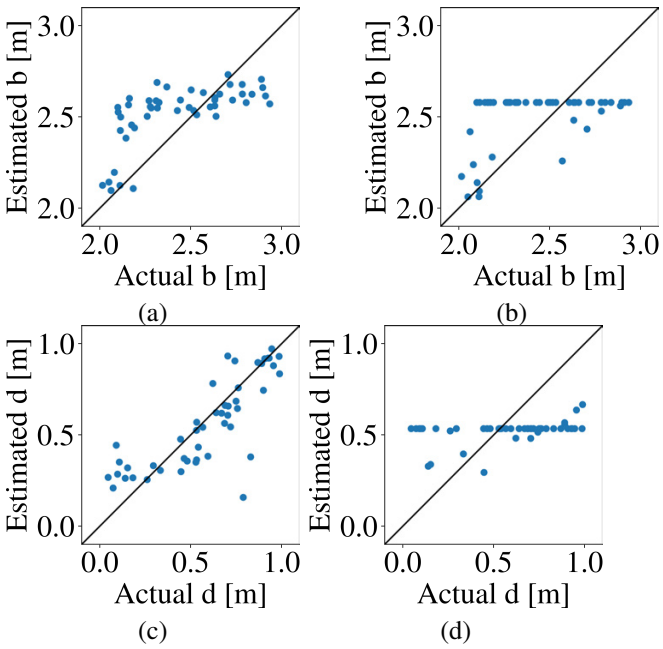


Fig. 9: Estimated vs. true values using (a,c) the set H with added noise and (b,d) the set G with added noise.

IV. Conclusion

The paper puts forward a framework that can enable the sensing and estimation of the features in the near field of an underwater swimmer based only on onboard measurements. This embodied sensing uses the spectral content of a Koopman operator associated with the sensor measurements (the observables). In the specific problem investigated in this paper, the singular vectors of the Koopman operator are used to train a CNN to estimate the location of an obstacle relative to an hydrofoil in an otherwise empty flow field. This method vastly outperforms a CNN that is trained to do the same estimation using direct pressure and velocity measurements. The spectral content of the Koopman operator captured in the ordered arrangement of the singular vectors in the input to the CNN contain features that are otherwise absent or hidden in an input matrix consisting of direct pressure measurements. This relatively high performance can also be expected to be true in the presence of sensor noise as shown by the results in this paper.

References

[1] M. S. Triantafyllou, G. D. Weymouth, and J. Miao. Biomimetic survival hydrodynamics and flow sensing. *Annual Review of Fluid Mechanics*, 48(1), 2016.

[2] M. Gazzola, M. Argentina, and L. Mahadevan. Gait and speed selection in slender inertial swimmers. *Proceedings of the National Academy of Sciences of the United States of America*, 112(13):3874–3879, 2015.

[3] I. A. Ijspeert. Biorobotics: Using robots to emulate and investigate agile locomotion. *Science*, 346(6206):196–203, 2014.

[4] E. Kelasidi, P. Liljeback, K. Y. Pettersen, and J. T. Gravdahl. Innovation in underwater robots: Biologically inspired swimming snake robots. *IEEE Robotics & Automation Magazine*, 23(1):44–62, 2016.

[5] T. J. Pitcher, B.L. Partridge, and C. S. Wardle. A blind fish can school. *Science*, 194(4268):963–965, 1976.

[6] H. Bleckmann and R. Zelick. Lateral line system of fish. *Integrative zoology*, 4(1):13–25, 2009.

[7] L. Sirovich. Turbulence and the dynamics of coherent structures. part 1: Coherent structures. *Quart. Appl. Math*, 45(3):561–571, 1987.

[8] P. J. Holmes G. Berkooz and J. L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, 25(1):539–575, 2003.

[9] R. Everson and L. Sirovich. Karhunen–loève procedure for gappy data. *J. Opt. Soc. Am. A*, 12(8):1657–1664, 1995.

[10] M. Damodaran T. Bui-Thanh and K. E. Willcox. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA Journal*, 42(8):1505–1516, 2004.

[11] K. E. Willcox. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Computers Fluids*, 35(2):208–226, 2006.

[12] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson. Spectral analysis of nonlinear flows. *Journal of fluid mechanics*, 641:115–127, 2009.

[13] P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

[14] E. M. Bollt Q. Li, F. Deitrich and I. G. Kevrekidis. Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the koopman operator. *CHAOS*, 27:103111, 2017.

[15] S. Otto and C. Rowley. Linearly recurrent autoencoder networks for learning dynamics. *SIAM Journal on Applied Dynamical Systems*, 18(1):558–593, 2019.

[16] J. N. Kutz K. Champion, B. Lusch and S. L. Brunton. Data-driven discovery of coordinates and governing equations. *PNAS*, 116(45):22445–22451, 2019.

[17] A. Yazdani M. Raissi and G. Karniadakis. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481):1026–1030, 2020.

[18] S. L. Brunton, B. R. Noack, and P. Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52:2020, 2019.

[19] K. Maeda J. L. Callaham and S. L. Brunton. Robust flow reconstruction from limited measurements via sparse representation. *Physical Review Fluids*, 4(2):103907, 2019.

[20] B. R. Noack S. Brunton and P. Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52(1):477–508, 2020.

[21] M. Alsalman, B. Colvert, and E. Kanso. Training bioinspired sensors to classify flows. *Bioinspiration & biomimetics*, 14(1):016009, 2018.

[22] B. Colvert, M. Alsalman, and E. Kanso. Classifying vortex wakes using neural networks. *Bioinspiration & biomimetics*, 13(2):025003, 2018.

[23] B. Pollard and P. Tallapragada. Sensing and classification of ambient vortex wake from the kinematics of a bioinspired swimming robot using neural networks. In *Proceedings of the Dynamic Systems Conference*, 2020.

[24] B. Pollard and P. Tallapragada. Learning hydrodynamic signatures through proprioceptive sensing by bioinspired swimmers. *Bioinspiration and Biomimetics*, 16(2):026014, 2021.

[25] J. M. Lidard, D. Goswami, D. Snyder, G. Sedky, A. R. Jones, and D. A. Paley. Data-driven estimation of the unsteady flowfield near an actuated airfoil. *Journal of Guidance, Control, and Dynamics*, 42(10):2279–2287, 2019.

[26] J. M. Lidard, D. Goswami, D. Snyder, G. Sedky, A. R. Jones, and D. A. Paley. Output feedback control for lift maximization of a pitching airfoil. *Journal of Guidance, Control, and Dynamics*, 44(3):587–594, 2021.

[27] B. Pollard and P. Tallapragada. Passive appendages improve the maneuverability of fish-like robots. *IEEE/ASME Transactions on Mechatronics*, 24(4):1586–1596, 2019.

[28] S. L. Brunton Y. Sun K. Duraisamy S. Bagheri S. T. Dawson K. Taira, M. S. Hemati and C. A. Yeh. Modal analysis of fluid flows: Applications and outlook. *AIAA Journal*, 58(3):998–1022, 2020.

[29] Z. Yao S. L. Brunton M. W. Mahoney N. B. Erichson, L. Mathelin and J. N. Kutz. Shallow neural networks for fluid flow reconstruction with limited sensors. *Proc. Royal Society. A*.

[30] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202.