2018

PASSIVE OBJECT DETECTION VIA AN ARTIFICIAL LATERAL LINE AND A 2-D VISCOUS FLOW MODEL

Jack H. Clark
University of Rhode Island, jack_clark@my.uri.edu

PASSIVE OBJECT DETECTION VIA AN ARTIFICIAL LATERAL LINE AND A 2D VISCOUS FLOW MODEL

BY

JACK H. CLARK

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
OCEAN ENGINEERING

UNIVERSITY OF RHODE ISLAND

2018

MASTER OF SCIENCE THESIS

OF

JACK H. CLARK

APPROVED:

Thesis Committee:

Major Professor	Jason Dahl
	Stephen Licht
	Christopher Roman
	Nasser H. Zawia DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND

2018

ABSTRACT

In the field of ocean exploration and marine vehicle design there is a heavy reliance on the use of acoustic and visual sensors for mapping, maneuvering, and obstacle avoidance. In low light regions, constricted spaces, or areas where water is turbid, however, passive systems similar to the lateral line of fish can be advantageous. Using a set of measurements like these, a digital twin could be constructed such that it represents the true physical system for use in mapping, control system design, or tracking the loading of a structure over time. For moving bodies in the presence of solid boundaries such as a vehicle operating at the ocean floor or a vessel in shallow waters, it is critically important to properly estimate the local boundary conditions for use with a digital twin.

The following study implements a viscous numerical simulation as a model for object identification. The presented methods include a classification schema via machine learning for a variety of wall shapes and sizes, as well as a general boundary estimation method using basis splines and an Unscented Kalman filter. At the cost of generality, the classifier is shown to be capable of identifying shape and size with relatively high accuracy especially as the number of available classes increases.

The basis spline method uses an Unscented Kalman filter and modeled pressure as a proxy to determine the locations of control points which govern boundary conditions. The method is shown to work for a single and multiple protrusions from a wall, with a weighted estimate approach developed which outperforms the standard formulation of the Unscented Filter when accounting for multiple measurement locations along the length of the foil body. Performance of the method as well as the sensitivity to the design of the system are discussed, and future work is presented.

ACKNOWLEDGMENTS

It is with great thanks and gratitude that I acknowledge my advising professor, Dr. Jason M. Dahl, for the guidance, support, and encouragement he has given me on this project. His dedication to helping myself and fellow students succeed both inside and outside the classroom is admirable, and without his willingness to assist me whenever I called upon him I don't know how I would have made it. He constantly pushed me to think about the problem outside of the context of the project, to how it could be applied, to practical thinking, and to the design of future systems such that I understood the true impact of the work we were doing together. I could not appreciate the fortune that I have had to work with Dr. Dahl more than I do, and I am eternally grateful for the opportunity I was given.

I would also like to acknowledge the continued support for this project from the Office of Naval Research (PM: Kelly Cooper) on grant #N30014-16-1-2968, without whom none of this would be possible.

Finally, I want to thank my peers Amin Mivehchi, Erdem Aktosun, Scott Hara, Bradely Brock, and Arielle De Souza for listening when I needed a sounding board, for all the working lunches, and for constantly telling me to hang in there and reminding me of the light at the end of the tunnel.

DEDICATION

I want to dedicate this thesis to my loving and supportive family and friends. Without your constant support and encouragement, I don't know how I would have made it through this.

To my parents, Lisa and Harry,

It is because of all the hard work and support that you both have provided me throughout my life that has brought me to this moment. You have always supported my goals and dreams, and done your best to assure that nothing held me back from achieving them. I could never express how grateful and thankful I am to both of you. You have truly made this possible.

To my sister, Kelly,

You have been my best friend and confidant through all of this. You have encouraged me to succeed at every step, and it has pushed me to be my best. Your constant belief in me has helped me to go so far, thank you for always being there when I needed you.

To Cony,

This has been one of the toughest experiences of my life, and you have been with me through all of it. You have supported me and loved me, and I cannot imagine having done this with anyone else by my side. Thank you so much. I look forward to our next adventure

TABLE OF CONTENTS

CHAPTER

Page

Page

APPENDIX

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

CHAPTER 1

Introduction

The ocean is a vast and largely unexplored region of the planet, and up until recently, there were very few types of vehicles that could be used to study these areas. However, in the last decade or so there has been a boom in marine exploration and marine vehicle design leading to a variety of different remotely controlled (ROV) and autonomous operated (AOV) vehicles such as gliders [5], floats [6], tethered crafts [7], etc... These vehicles employ a variety of sensors and systems for measuring their surrounding environment. For ROVs and AOVs there are different types of measurements to determine the location and general surroundings of the vehicle such as cameras, lasers, acoustic signals, and radar. These work well in many situations, however they become limited in regions where there is low light, reflective surfaces are jagged, or the water is turbid. Additionally, typical sensing systems require an active stimulation for measuring i.e. cameras require light to reflect off an object, acoustic systems require an acoustic pulse to reflect of an object. For this reason, there is a desire to investigate systems that can utilize passive measurements such as pressure or local flow velocities for use with these types of vehicles.

In addition, there has been a surge of interest in the development of digital twins for ocean vehicles. A digital twin is a model representation of a physical system which properly captures the physical surroundings and the local dynamics in order to provide an artificial representation of a system [8]. Given proper design and a set of inputs, a digital twin can give estimates of product life-cycle, fatigue life, fatigue points, and even optimize design [8, 9]. Digital twins have been implemented for product design and manufacturing [10, 11], for vehicle "health" monitoring

[12], improving fuel economy [13], and floating production system monitoring [14]. While digital twins have been combined with dynamic systems in the past, they have been predominantly used for some type of optimization or fatigue life analysis.

Digital twins can be particularly useful for marine vehicles and control systems as they may be used in real time control or decision making. In order to inform the model of the physical surroundings, measurements must be assimilated into the twin via some filtering or ensemble method. This can be a particular challenge in situations such as a streamlined body passing by an object in fluid. In this flow field, the boundary conditions are constantly changing as the body moves through the medium, and those that represent the object are unknown. This becomes even more complex due to the presence of a fluid wake, which has memory and changes in time.

The following work describes a method for determining general boundary conditions in a simple 2D viscous flow model of a wing passing by an object on a wall. The presented method is a first step towards realizing a real time computational model integrated with the operation of a moving body in a fluid providing updated boundary conditions. The estimated boundary conditions provide a model which represents the local surroundings and could be implemented with a digital twin.

1.1 Biological Analog

Measurements of pressure on the surface of a foil were used as the input for determining local boundary conditions. A biological analog for this type of measurement is the mechanosensory lateral line of fishes and aquatic amphibians. The lateral line is a system composed of sensory organs that run along the head and body (Figure 1) which can sense changes in pressure and flow in the surrounding fluid [15]. Fish use this system for a variety of complex behavior such as schooling, predator avoidance, prey detection, communication, object detection and avoidance, and rheotaxis [16]. Due to the nature of adaptation across diverse environments, the lateral line has evolved for a variety of different tasks [3], including detecting flow conditions [17], encoding a Karman Vortex Street [1], and object identification [18].

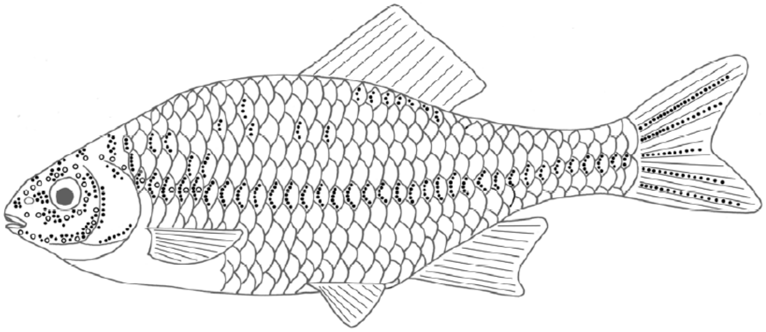


Figure 1: A drawing showing the distribution of superficial neuromasts (dots) and pores of the lateral line canal (circles) in the bitterling, *Rhodeus sericeus amarus* (Cyprinidae) [1].

The system is comprised of a network of sensors called neuromasts which exist both on the skin and inside a series of porous canals just below the skin (Figure 2). A neuromast is a bundle of hair cells covered by a gelatinous mass called a cupula, which provides an interface between the hair cells and surrounding fluid [19]. The bundle can contain anywhere from hundreds to thousands of hairs each with a ciliary bundle at the apical surface of the cell, whose function is closely related to that of hair cells found in the auditory systems of vertebrates. The bundles shown in Figure 3 are composed of stereocilia organized in order of their height with the tallest next to an elongated kinocilium [1, 3].

There are two types of neuromasts: superficial neuromasts (SN) and canal neuromasts (CN) as seen in Figure 3. SNs are smaller receptors on the surface of the body that are sensitive to changes in fluid velocity. CNs, however, are larger pressure-gradient-sensitive receptors which appear to be able to respond to more rapid changes [20]. The difference in these forms of stimuli can be seen in Figure 4,

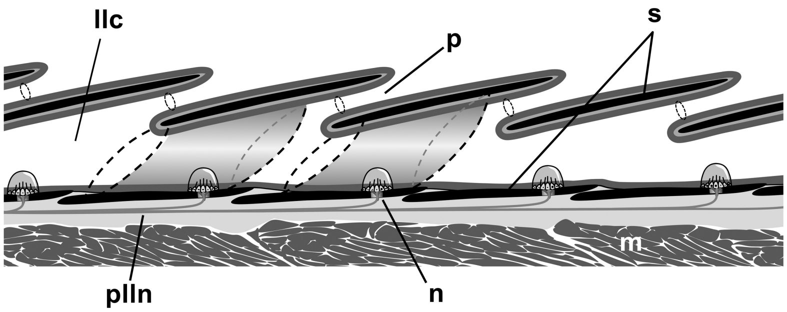


Figure 2: A diagram of the Lateral Line Canal (llc) on the trunk, highlighting the canal neuromasts (n) and the posterior lateral line nerve (plln). Also shown are the pores, or openings, from the canal to the surrounding fluid (p), the scales of the fish (s), and the underlying muscular tissue (m) [2]

where the SN is stimulated by an external flow of fluid while the CN is stimulated by pressure changes inside the canal due to movement of fluid along the pores of the canal.

Studies of behavior in fish have shown that blind cave fish are capable of detecting the presence of new obstacles in their environment and their ability to recognize both the size and shape of the object [21]. It has also been shown that the blind cave fish can be trained to make decisions based on object shape, they will make use of their lateral line canal (llc) by passing by the object prior to decision making [22]. The intriguing functionality of this sensory organ has lead to research in the field of artificial lateral line sensing for a variety of marine applications.

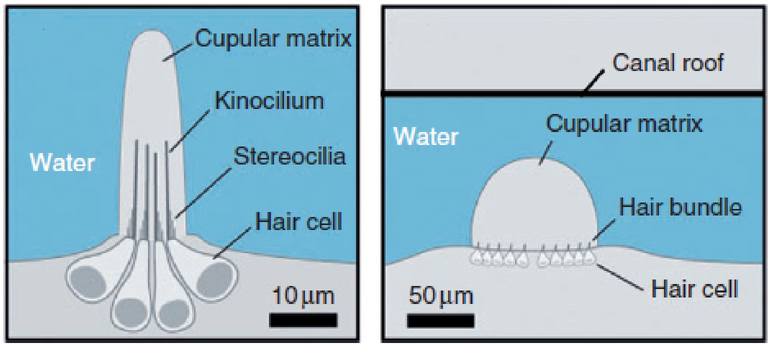


Figure 3: Diagrams of a superficial neuromast (left) and the canal neuromast (right) [3]

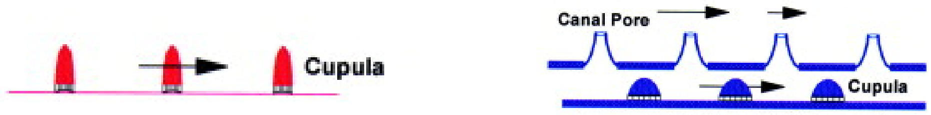


Figure 4: Diagram illustrating the difference in stimuli between the superficial neuromast (left) and the canal neuromast (right) [4]

1.2 Previous Work on Artificial Lateral Lines

The field of artificial lateral line sensing and object recognition has been approached from a variety of angles from both the side of bio-mimicry and bio-inspiration. Much of this research has been focused on the design of novel pressure and velocity sensors that mimic the lateral line system [23, 24, 25] with a variety of methods developed for interpreting these measurements. Sensing systems such as these could be used for mapping, navigation, or control system design for marine vehicles especially in turbid environments where traditional sensing systems become compromised.

1.2.1 Artificial Lateral Lines Sensor Development

Due to the types of stimuli to which it depends, engineers and scientists often considered the lateral line system analogous to an array of pressure sensors. Fernandez et al. [26] developed a set of small pressure sensors that could contour to a wide variety of marine vehicles which could mimic this system. They tested a strain-gage pressure sensor with a diameter near 1 mm, and spaced them a few millimeters apart on a sensing body to mimic the lateral line. The resolution of the sensors was on the order of 1 Pa. Systems such as these have been built for a wide variety of investigations, including the detection of ground effect [27], hydrodynamic imaging [28], and object tracking [20].

Venturelli et al. [29] demonstrated that artificial lateral lines on fish-like robots could be used to distinguish between uniform fluid flows and a Karman vortex street. The Karman vortex street was distinguished from the uniform flow via use of the turbulence intensity on the surface of the body and a frequency spectrum analysis (FFT) to find the number of sensors with the same dominant frequency. Pressure differences across the body were used to determine orientation and position of the body, and cross-correlation among the sensors was used to find the vortex traveling speed. Using this information, an estimate of wake wavelength was also determined based on vortex traveling speed and shedding frequency [29].

1.2.2 Object Detection and Identification

The problem of object detection is of particular interest for marine vehicles in regions where the ocean floor is close. It was shown by Fernandez et al. [30] and Maertens et al. [31] that object detection and tracking could be accomplished using pressure sensors in both passive and active systems. Bouffanais et al. [32] showed that location, size, and general shape of an object could be determined from embedded pressure sensors. In their experiments, a potential flow panel method was used to model a foil and a generalized transform was used to describe a shape near the foil. The shape description allowed for simple definitions of characteristics like size, location, and orientation. Through inversion of the pressure measurements via an unscented Kalman Filter it was possible to separately determine these characteristics, though it was determined that the sensing body must be sufficiently close to the object in order to properly characterize the object (within one body length).

Maertens et al. [31] applied this method of detection to a variety of object shapes and multiple bodies. In these experiments, a NACA 0018 foil body with embedded pressure sensors was dragged past a stationary object and the method described by Bouffanais et al. [32] was used to model object characteristics. The method of detection was validated for regimes which closely match potential flow approximations, however estimation of shape parameters can break down when flow separation causes significant deviations in the pressure determined from potential flow models. Additionally, the definition of object shapes in the generalized transform limited object recognition to a set of predefined shapes due to the nature of the parameterization method.

It has been demonstrated that viscous effects are critically important and influence physical lateral line systems on both the boundary layer and receptor scale [33]. Maertens and Triantafyllou [34] established that in the presence of separation, the shape information derived from pressure sensing may be improved through inclusion of viscous effects. In these experiments it was first determined through use of viscous simulations and potential flow models that the boundary layer on the surface on a moving fish is significantly influenced by unsteady viscous effects. Further, through linear stability analysis of the boundary layer of a cylindrical body, a correction was applied to potential flow modeled pressures based on the displacement thickness of the boundary layer. It was also seen in these experiments that the boundary layer can act as a signal amplifier, providing improved estimates of object characteristics. While this provided evidence that viscous effects are important in object recognition, the method was limited due to the fact that viscous information was determined a priori through learning algorithms and thus could not be applied in real time.

Further, Boulogne et al. [35] applied the techniques of machine learning and neural networking for object localization using an artificial lateral line. They implemented a variety of machine learning algorithms to simulated pressure time histories of a cylinder moving past a stationary lateral line array. The cylinder was allowed to move through an x,y-plane at an angle relative to the sensor array, and the corresponding pressure from a potential flow model was used to train the network to determine the horizontal and vertical position of the cylinder relative to the array at any point in time. It was shown here that it was possible to determine the position of the cylinder relative to the sensing array with low euclidean error, however the exclusion of viscous effects and the use of a stationary sensing system limited the applicability of this method to a moving body. As it is known that the viscous boundary layer is critically important to the function of the lateral line of fish [33], it is necessary to use a more accurate model of the fluid in order to make this method more applicable in a marine setting.

1.2.3 Statement of Purpose

To date, no experiments have been done to determine a general object identification method the uses a viscous fluid model. For the development of a general method, the problem of a streamlined body passing by a protrusion from a flat wall was considered. Akin to a fish swimming by an object, measurements of pressure on the surface of the body were used in conjuncture with the developed method to determine characteristics of the protrusion. The purpose of this investigation is to improve upon previous work in two key areas. First, a fully viscous fluid model will be applied to directly include viscous effects in an object detection algorithm. Second, to characterize an object's general shape, two methods will be investigated for object detection. Object recognition is performed using classification via machine learning as well as shape definitions via standard spline definitions coupled with an Unscented Kalman filter. These modifications use a viscous flow model for boundary identification to enable boundary updating in real-time digital twin systems.

1.3 Document Organization

Chapter 2 introduces the 2-D viscous simulation used as the model for the object recognition method. This is both an introduction of the simulation and an explanation of the implementation of modifiable boundary conditions.

Chapter 3 provides a method of object recognition using machine learning and neural networking.

Chapter 4 demonstrates a more general method of object recognition using an Unscented Kalman filter.

Chapter 5 provides an investigation and discussion of the method as well as an analysis of the results.

Chapter 6 and Chapter 7 provide conclusions and recommendations for future work.

CHAPTER 2

Viscous Model

While potential flow models are known to provide reasonable solutions, the response can diverge from the behavior seen in physical systems when there is turbulence in the fluid or large separation regions form on the body. This can cause object detection algorithms to fail when readings of pressure are made in the wake of a moving body. For this reason, this study looked to improve upon previous methods of object estimation via implementation of a viscous fluid model.

To develop a method for estimating the shape of an object near a wall, the general setup of the problem is first defined. For this problem, a streamlined body (NACA 0012 wing section) at zero angle of attack with forward speed, U, passes by a feature protruding from a flat wall. Figure 5 shows the foil body near an example protrusion. Measurements of pressure are assumed to be taken on the surface of the foil body (marked by red circles in Figure 5) at evenly spaced locations. Using a viscous simulation of the fluid field as a forward model, pressure is used to infer characteristics of the wall as the foil moves in time. Observing the pressure in the fluid at discrete values of time, k, demonstrates the formation of the wake over time and how it is affected by the presence of an object on a wall. Figure 6 shows the influence of a wall feature on the pressure in the fluid at three instances in time in LilyPad. Here it can be seen that the change in pressure in the fluid is gradual when approaching and moving away from the wall feature, with an increase in the magnitude of the pressure when the body is in the proximity of the wall feature. This demonstrates that the formation of the wake is gradual as the body moves through the fluid, and that the presence of the body causes a measurable change in the pressure. In the development of the algorithm, all measurements of pressure were generated through simulations, and no physical measurements were used, although the algorithm will be tested with physical measurements in the future.

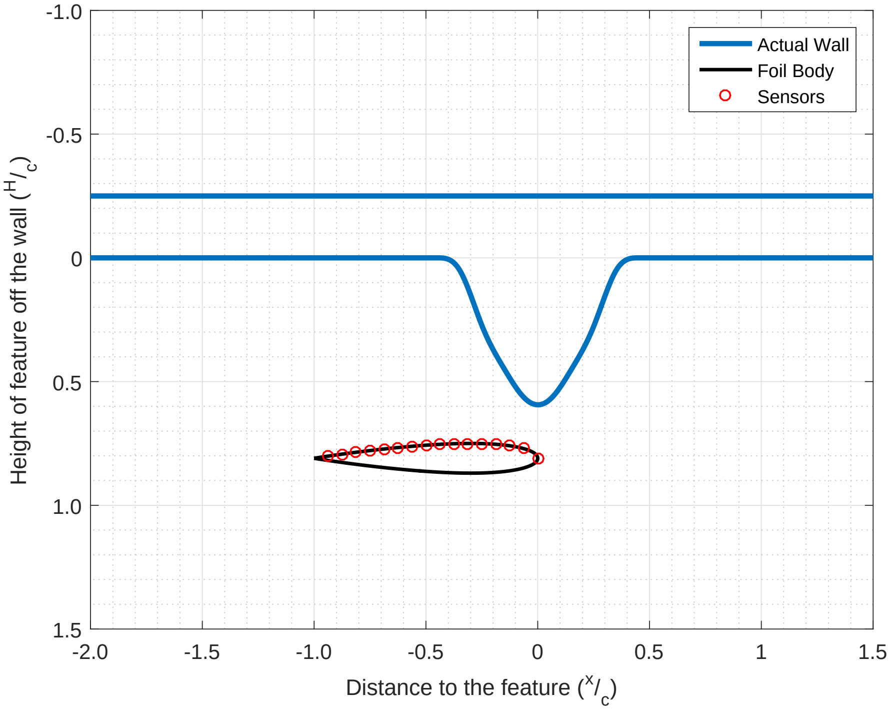
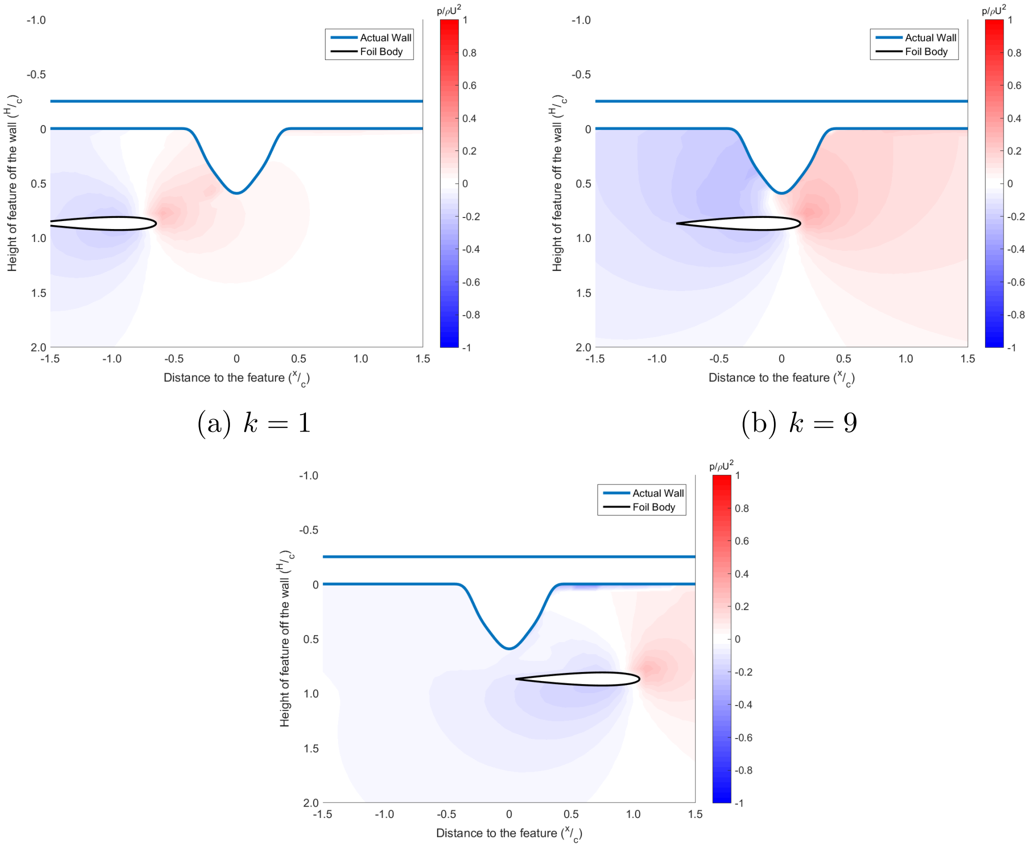


Figure 5: A diagram of the simulation showing the distribution of pressure sensors.

2.1 Simulation Definitions

The numerical model used to represent the system was an open source, 2-D, Cartesian-Grid, Navier-Stokes solver named LilyPad [36]. Lilypad is a numerical tool designed to perform complex fluid-structure simulations at low computational cost. It is of particular interest to researchers as it provides immediate visual feedback and is easily modified. It implements the boundary data immersion method (BDIM) [37], an immersed boundary technique which uses kernal functions to blend field equations and boundary conditions for fluids and solids near solid boundaries. BDIM has been demonstrated to achieve second order accuracy in force estimates and has been validated for a variety of fluid-structure interaction



(a) $k = 1$

(b) $k = 9$

(c) $k = 18$

Figure 6: The pressure in the fluid as the foil body passes by an object on a wall at three discrete points in time. The pressure from LilyPad is shown with contours of color described by the color bar to the right of each plot.

problems [34].

Typically, it is necessary to test the convergence and validation of a simulation in order to determine what characteristics and scales must be matched to properly capture the physics of the problem. In this investigation, however, it was of interest to develop an algorithm that could determine the conditions of the system based on a set of measurements from a simulated system. Due to this, it was not critically important that the model capture the exact physics of the problem. For this reason, simulations were under resolved in space and resolved enough in time to provide a stable measurement of pressure such that the computational time of each simulation was reduced. On the contrary, implementing this method with physical measurements would require that the model be sufficiently resolved in both space and time, such that the flow field is representative of the true physical system.

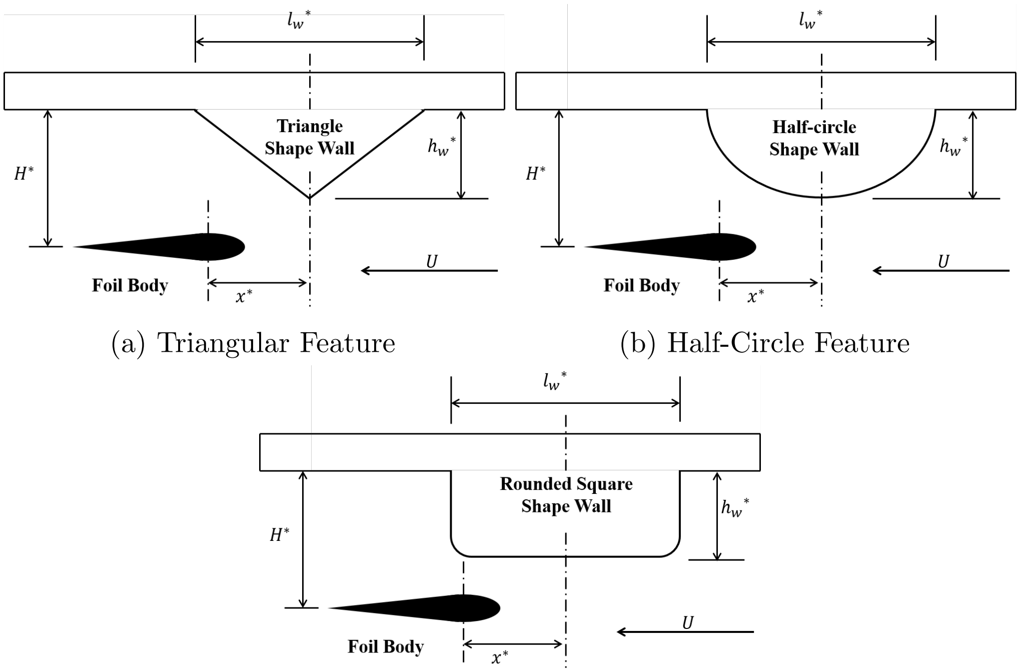
The resolution of the Cartesian grid was defined with 2^7 or 128 computational nodes along the horizontal and vertical axis of the domain. A long wall was placed in the simulation in the presence of a NACA 0012 foil with a body length of 10 grid points. Proper resolution of the flow field for this problem requires approximately 20 grid points along the foil, and a grid size of 2^9 or 2^{10}, however this reduced domain was used to help speed up computational processing in the development of the estimation algorithm.

The simulation was designed with a stationary foil and a wall with features that moved past the body at a velocity equal to the free stream velocity in the fluid, U. This is equivalent to the body moving through a still fluid, and avoided boundary layers forming on the wall from fluid motion. Due to the resolution on the body, measurement locations were chosen at 16 unique points on the surface of the foil (shown in Figure 5) and pressure was determined as a function of time via interpolation from the nearest computational grid point.

2.2 Object Shape

In the development of the estimation methods, several standard shapes are also defined in order to investigate the use of pressure as an identifier of object shape. Three shapes were investigated initially: a triangle, a half-circle, and a rectangle (shown in Figure 7). Length scales in the simulations were determined in non-dimensional units of chord lengths of the foil such that the height h_w^*, length l_w^*, distance from the wall H^*, and distance to the center of the feature x_w^* were all relative to the size of the body. See Table 1 for further explanation of these parameters.

Here it was hypothesized that for any given feature of a specific size and shape, there would be a unique pressure signature along the length of the foil. This can be seen in Figure 8 and Figure 9 where pressure signatures for the three shapes of interest with varying height and length are presented. Figure 8 demonstrates the effect of varying the height of the feature on the measured pressure while Figure 9 shows the effect of varying length. All simulations were performed at a constant value for H^*, and the pressure is presented as a spatial history relative to the location of the foil as it moves past the feature.



(a) Triangular Feature

(b) Half-Circle Feature

(c) Rounded Square Feature

Figure 7: A diagram of the design of each wall shape for simulations. $h_w^*, l_w^*, H^*,$ and x^* refer to the height and length of the feature, the distance between the foil and the wall, and the distance between the foil and the center of the feature, all non-dimensionalized by the chord length of the foil.

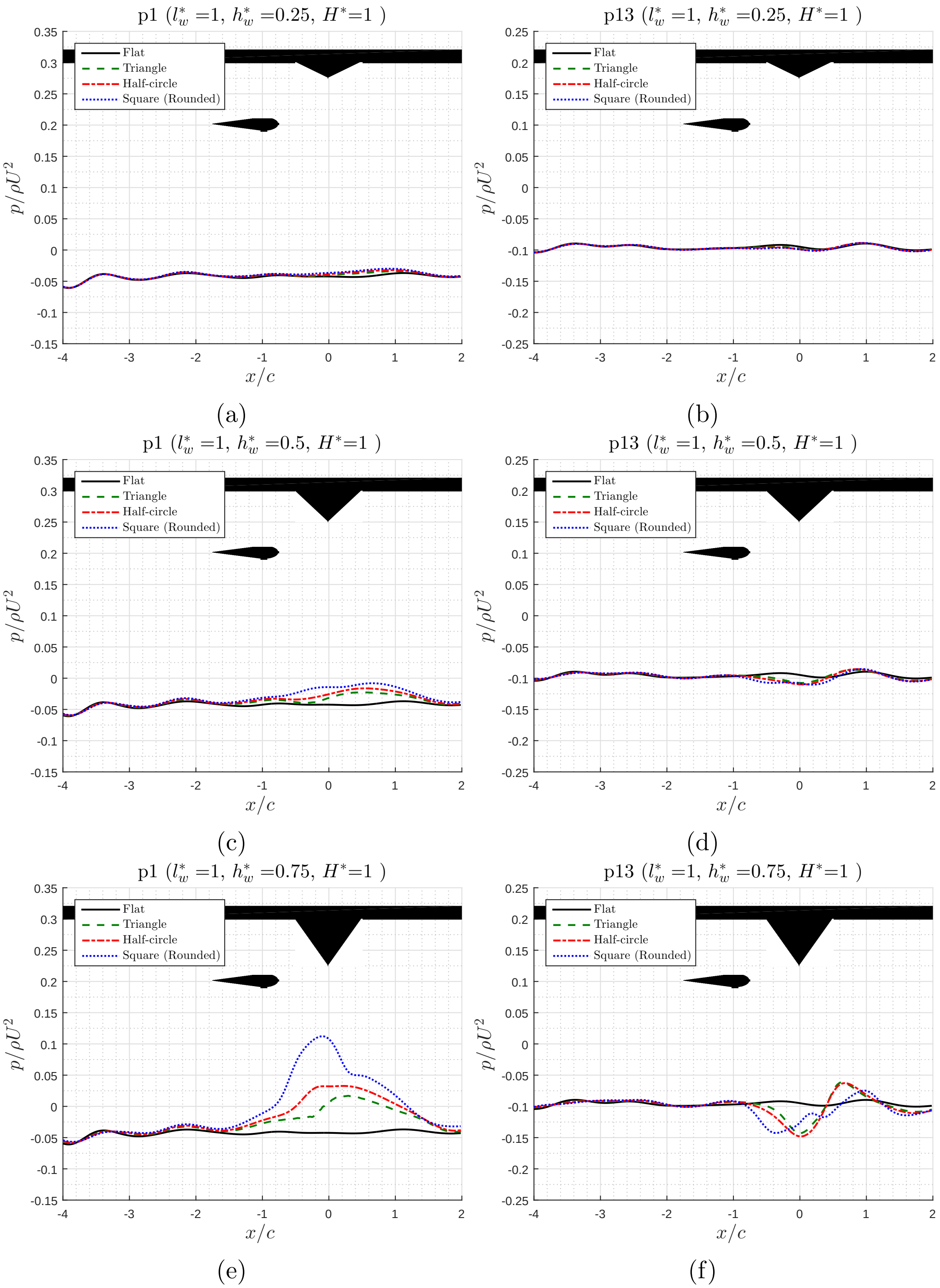
From Figure 8 it is clear that as the height of the feature increases, the differences in the shapes of the pressure signatures becomes more pronounced. Further it can be seen that while the front of the foil (left column of Figure 8)

Table 1: List of geometric parameters of the wall feature passing by the foil body.

Length of the bump (length-to-chord)	l_w^*	Also: l/c or L/c
Height of the bump (height-to-chord)	h_w^*	Also: h/c or H/c
Distance between the foil and the flat wall	H^*	Also: H
Distance between the foil and the center of the feature	x^*	Also: x/c

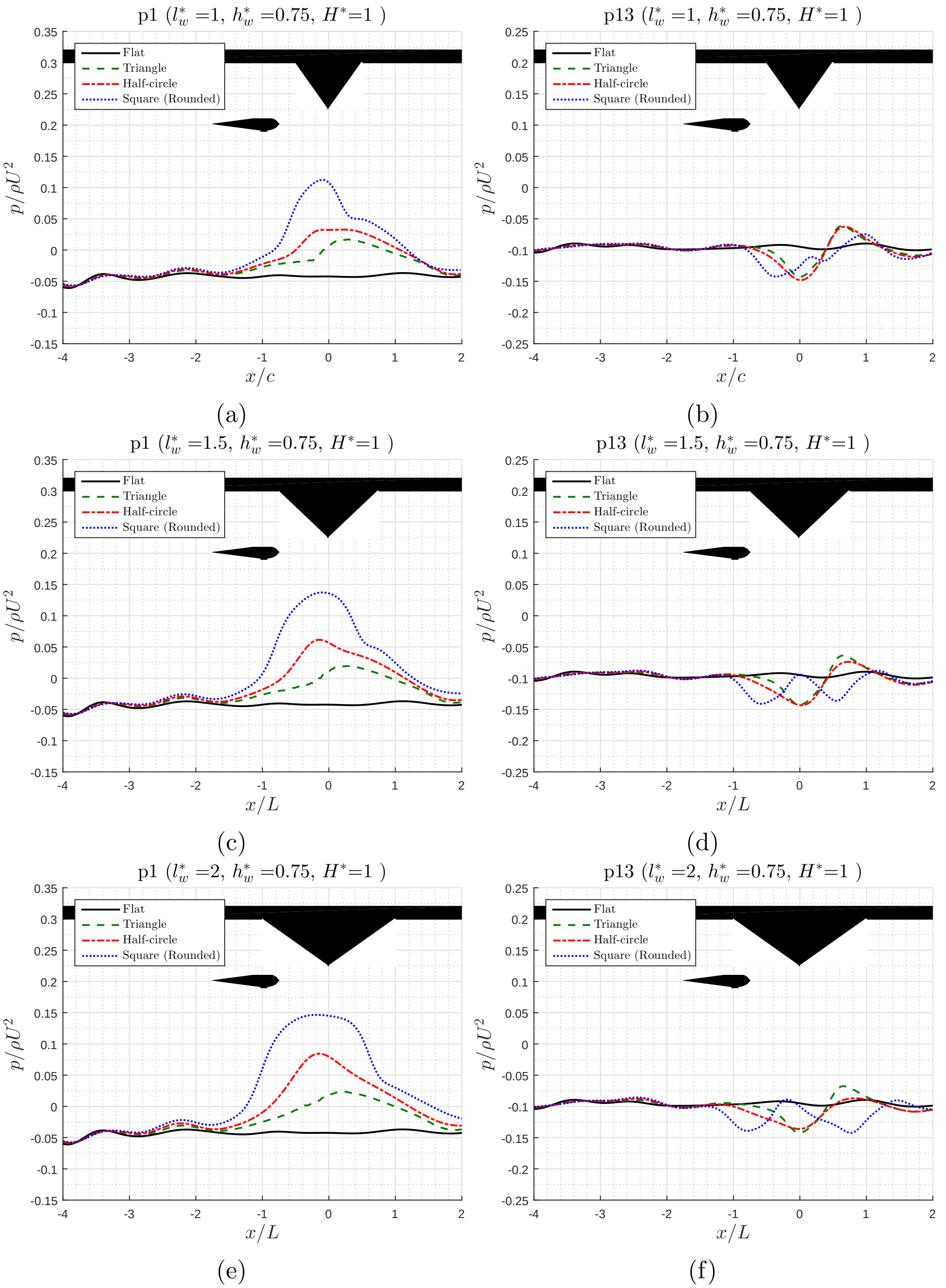
shows relatively unique pressure signatures, the shapes of the pressures seen in the wake of the foil tend to be much closer in trend and shape (right column of Figure 8). Comparing to Figure 9, it can clearly be seen that the length of the feature is critically important for distinguishing between shapes. Observing both columns in Figure 9, it can be seen that the pressure signatures begin to widen and show differences between one another.

The noticeable difference in these pressure signatures shows that pressure in a wake contains unique information about the object that created the wake. It is expected that similarities in time histories could make distinguishing shapes difficult, so pressure signatures on the leading edge and wake of the foil are used to identify unique information about features. Using this information, a neural network is developed to investigate shape recognition similar to the experiment by Boulogne et al. [35].



(a) (b) (c) (d) (e) (f)

Figure 8: Pressure on the surface of the foil for various shapes with equal length and varying height. The foil body moves from left to right, where $^{x}/_{c} = 0$ refers to the point where the foil body passes the center of the feature. Measurements are given at the front of the foil (left column) and in the wake (right column).



(a) (b) (c) (d) (e) (f)

Figure 9: Pressure on the surface of the foil for various shapes with equal height and varying length. The foil body moves from left to right, where $^{x}/_{c} = 0$ refers to the point where the foil body passes the center of the feature. Measurements are given at the front of the foil (left column) and in the wake (right column).

CHAPTER 3

Classification with Machine Learning

Machine learning can provide subtle solutions, and make connections that otherwise would be impossible to model [38]. This can be both a blessing and a curse, as neural networks often provide accurate answers, but require massive training sets to give sufficiently generalized solutions. Using machine learning allows for the design of a classifier that can make decisions about what types of features exist in the local flow field based on pressure time histories. There exist two major processing types in machine learning that could be applied to this type of data: supervised learning, and unsupervised learning. Supervised learning is used when an answer is known and a set of inputs and corresponding outputs can be compiled such that the network can determine a relationship between inputs and outputs. Unsupervised learning is typically used when recognizable patterns in the data are not already known and rules must be developed based solely on observed data [38].

Supervised learning is used in neural networking to train predictors based on known inputs and outputs. Under the category of supervised learning there exists a number of training models that can be implemented based on the structure of the data. Three important methods that can be used in a classification learner are linear/non-linear discriminant learning, support vector machine (SVM) learning, and k-Nearest Neighbors (kNN) learning. The discriminant learning method and the SVM share similar properties, however a discriminant learner attempts to separate inputs based on multi-degree lines and a SVM method seperates inputs based on multi-degree hyperplanes [39]. Both of these methods create separations between clusters to judge where inputs fall based on their distance to each separation, and uses these to make predictions about unknown inputs. The kNN method is different due to the fact that it bases clusters of inputs based on k similarities seen in the data set, meaning that as k increase so does the amount of overlap in classes leading to a decrease in accuracy [39].

Due to the nature of this problem it was determined that supervised learning was the appropriate avenue for designing a classifier, with pressure time histories as input for determining wall shapes.

3.1 Training

For classification, a significant amount of training data is necessary in order to make accurate predictions. In order to define a simple classifier, the distance from the wall to the foil, H^*, was fixed for all simulations such that the classifier was only given pressure time histories from features of varying height and length. The non-dimensional distance between the foil and the wall, H^* was set at a constant value of 1.33 for all simulations. A range of non-dimensional feature height, h_w^*, was chosen from 0.05 to 1.2 with steps of 0.05, and a range of non-dimensional length, l_w^* was chosen form 0.05 to 3.0 with steps of 0.05.

This generated pressure time histories for 1445 combinations of feature height and length for each predefined shape, a total of 5760 simulations including flat wall references. Half of this data, based on the height of the feature, was set aside for training the classifier and half for a secondary validation after training was completed. This separation of data was done to assure that data used for the secondary validation had not been used to train the classifier, and thus the network could be tested with completely unknown inputs. The data was separated into a Matlab table, similar to Table 2, with columns representing pressure time histories from multiple locations along the length of the foil as inputs to the system and the final column corresponding to the known class. The table was designed such each row corresponded to a particular simulation run, meaning that in the training set there was a total of 2880 rows. This table was loaded into the classification learner toolbox in Matlab using an internal cross-validation method to avoid over-fitting the classifier. Cross-validation partitions the training data into a set number of “folds”, and the accuracy of the classifier is determined over each fold before being globally compared. It was decided to use 30 folds, and to simultaneously train the classifier using the three methods mentioned above, such that the accuracy of each learning style could be compared.

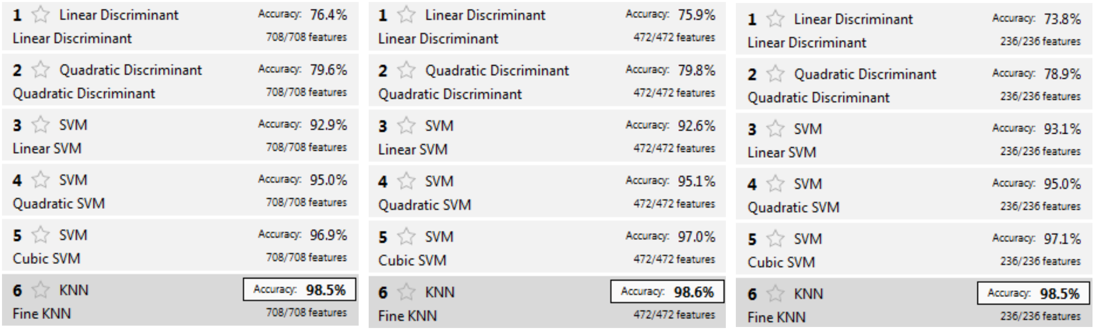
Table 2: An example of the style of the input for the classifier.

P_1	P_2	P_4	$\dots$	P_{14}	P_{15}	P_{16}	**Class**
Time history of pressure							Shape/Flat

3.2 Triangle Versus Flat Wall

For the development of a classifier, it is critically important that the training sets used provide information sufficient to distinguish between known classes. Six measurements of pressure along the length of the foil body, three near the leading edge and three in the wake, were used to provide information about the protrusions class. The data was then separated such that only simulations pertaining to flat walls or triangles were input into the training table, in order to determine the classifiers ability to distinguish a single feature from a flat wall.

A number of classifiers were trained using decrement methods, SVM methods, and a kNN method to determine which style of classification fit the data best. It was then relevant to remove the number of inputs until a reduction in accuracy was noticed. This was done by re-training the networks with fewer time histories and observing the accuracy of prediction. As can be seen below in Figure 10, the prediction ability of each model was relatively unaffected as the number of time histories was reduced.



	Model	Accuracy	Features
1 ☆	Linear Discriminant Linear Discriminant	Accuracy: 76.4%	708/708 features
2 ☆	Quadratic Discriminant Quadratic Discriminant	Accuracy: 79.6%	708/708 features
3 ☆	SVM Linear SVM	Accuracy: 92.9%	708/708 features
4 ☆	SVM Quadratic SVM	Accuracy: 95.0%	708/708 features
5 ☆	SVM Cubic SVM	Accuracy: 96.9%	708/708 features
6 ☆	KNN Fine KNN	Accuracy: 98.5%	708/708 features

(a) All six points of pressure

	Model	Accuracy	Features
1 ☆	Linear Discriminant Linear Discriminant	Accuracy: 75.9%	472/472 features
2 ☆	Quadratic Discriminant Quadratic Discriminant	Accuracy: 79.8%	472/472 features
3 ☆	SVM Linear SVM	Accuracy: 92.6%	472/472 features
4 ☆	SVM Quadratic SVM	Accuracy: 95.1%	472/472 features
5 ☆	SVM Cubic SVM	Accuracy: 97.0%	472/472 features
6 ☆	KNN Fine KNN	Accuracy: 98.6%	472/472 features

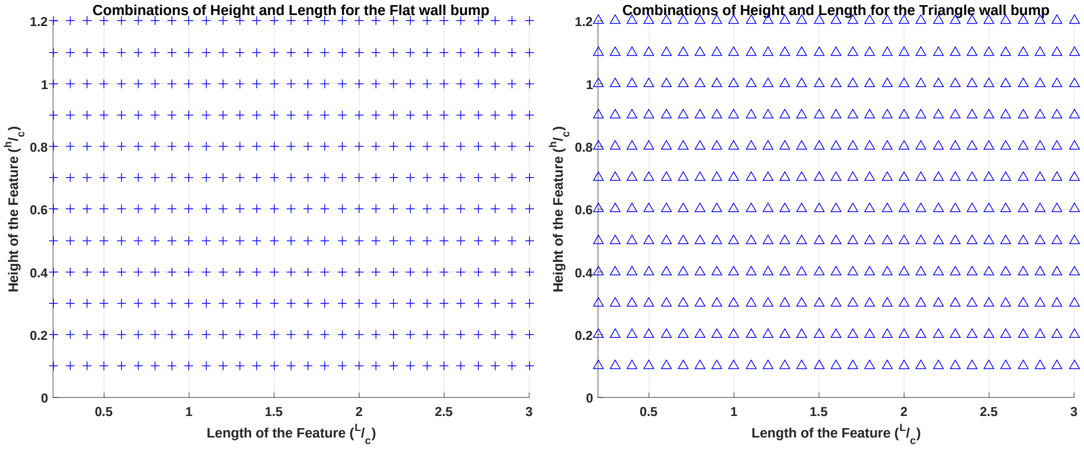
(b) Points 3, 4, 13, 14

	Model	Accuracy	Features
1 ☆	Linear Discriminant Linear Discriminant	Accuracy: 73.8%	236/236 features
2 ☆	Quadratic Discriminant Quadratic Discriminant	Accuracy: 78.9%	236/236 features
3 ☆	SVM Linear SVM	Accuracy: 93.1%	236/236 features
4 ☆	SVM Quadratic SVM	Accuracy: 95.0%	236/236 features
5 ☆	SVM Cubic SVM	Accuracy: 97.1%	236/236 features
6 ☆	KNN Fine KNN	Accuracy: 98.5%	236/236 features

(c) Points 4 and 13

Figure 10: How using different numbers of points changes the accuracy of the classifier for triangle vs flat wall.

It is clear that the kNN network model (with k set to 1) fit the data best due to its increase in accuracy over other classification schema. Due to this, the kNN model trained using points 4 and 14 on the foil was exported and a second validation step with the other set of data was done. This can be seen below in Figure 11, where Figure 11a corresponds to flat wall cases and Figure 11b shows triangle feature cases.



(a)

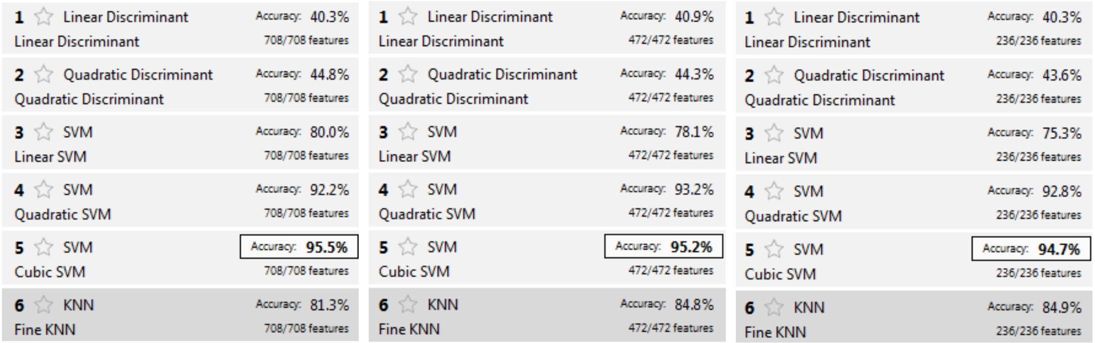
(b)

Figure 11: Personal validation of the classifier using both flat wall and triangle feature pressure signatures. Plus signs signify flat walls, triangles signify triangle features. A blue markers represent a correct classification.

In Figure 11 it can be seen that classification between a flat wall and a triangular feature protruding from a wall is possible, and easily done by the kNN network. Even the smallest triangular feature was distinguishable from a flat wall due to the differences in magnitude seen in Figure 8. For this reason, it was determined that complexity could be added to the network to test it's ability when more class options are available.

3.3 All Shapes

The classifier is next defined using four available classifications: flat wall, triangular wall, circular wall, and a rounded square wall. This was done in a similar manner as before where 1445 sets of time histories were generated for each shape. Again, half of the data was used for training and half of the data was set aside for later validation of the exported model. The same six points on the foil were used as time histories in the classification method. Fitting of the data to a variety of classification methods is seen in Figure 12 for different combinations of pressure time history locations.



1 ☆ Linear Discriminant Accuracy: 40.3%
Linear Discriminant 708/708 features

2 ☆ Quadratic Discriminant Accuracy: 44.8%
Quadratic Discriminant 708/708 features

3 ☆ SVM Accuracy: 80.0%
Linear SVM 708/708 features

4 ☆ SVM Accuracy: 92.2%
Quadratic SVM 708/708 features

5 ☆ SVM Accuracy: 95.5%
Cubic SVM 708/708 features

6 ☆ KNN Accuracy: 81.3%
Fine KNN 708/708 features

(a) All six points of pressure

1 ☆ Linear Discriminant Accuracy: 40.9%
Linear Discriminant 472/472 features

2 ☆ Quadratic Discriminant Accuracy: 44.3%
Quadratic Discriminant 472/472 features

3 ☆ SVM Accuracy: 78.1%
Linear SVM 472/472 features

4 ☆ SVM Accuracy: 93.2%
Quadratic SVM 472/472 features

5 ☆ SVM Accuracy: 95.2%
Cubic SVM 472/472 features

6 ☆ KNN Accuracy: 84.8%
Fine KNN 472/472 features

(b) Points 3, 4, 13, 14

1 ☆ Linear Discriminant Accuracy: 40.3%
Linear Discriminant 236/236 features

2 ☆ Quadratic Discriminant Accuracy: 43.6%
Quadratic Discriminant 236/236 features

3 ☆ SVM Accuracy: 75.3%
Linear SVM 236/236 features

4 ☆ SVM Accuracy: 92.8%
Quadratic SVM 236/236 features

5 ☆ SVM Accuracy: 94.7%
Cubic SVM 236/236 features

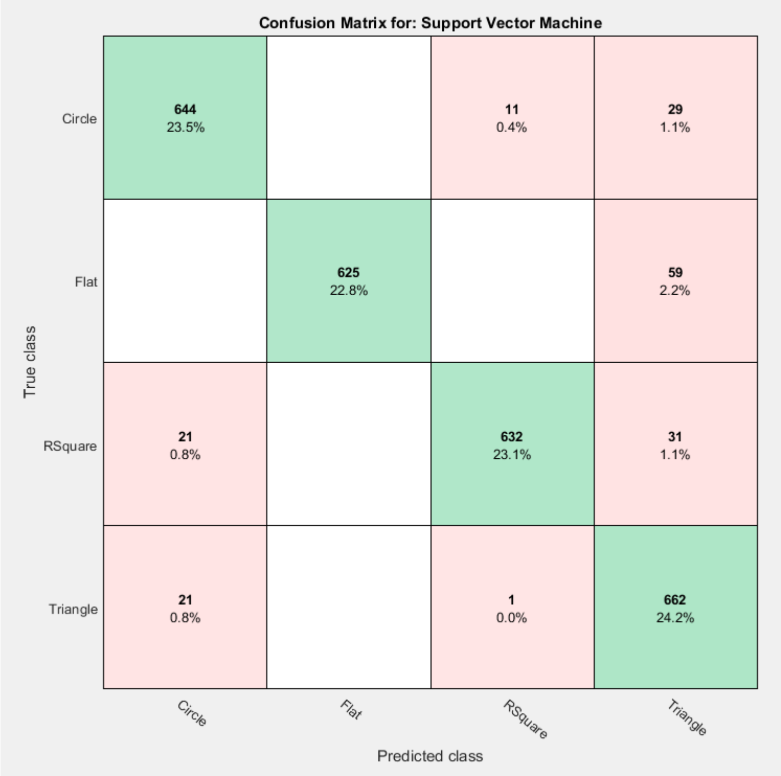
6 ☆ KNN Accuracy: 84.9%
Fine KNN 236/236 features

(c) Points 4 and 13

Figure 12: How using different numbers of points changes the accuracy of the classifier for all wall features of interest.

It can again be seen that as long as a point at the front and a point in the wake are used, only two points are needed for classification. Interestingly, the cubic SVM outperforms the kNN model when more shapes are used, which can be seen by comparing Figure 12 and Figure 10. Before, the kNN only outperformed the SVM models by a 1-2%, but when the problem became more complex, the cubic SVM was 10% better at estimating the correct shape.

Further information can be gleaned about the performance of this classifier by observing the confusion matrix from the machine learning toolbox (Figure 13). Here it can be seen that 2.2% of the flat wall features in the training of the neural network were misclassified as triangular walls, meaning that a protrusion was predicted when none existed. Further, the majority of misclassification was accidentally classifying a feature as a triangle when it was not. Most important to note is that a wall feature was never misclassified as a flat wall, meaning that while there may be confusion in shape, there is no confusion about the fact that a protrusion existed.

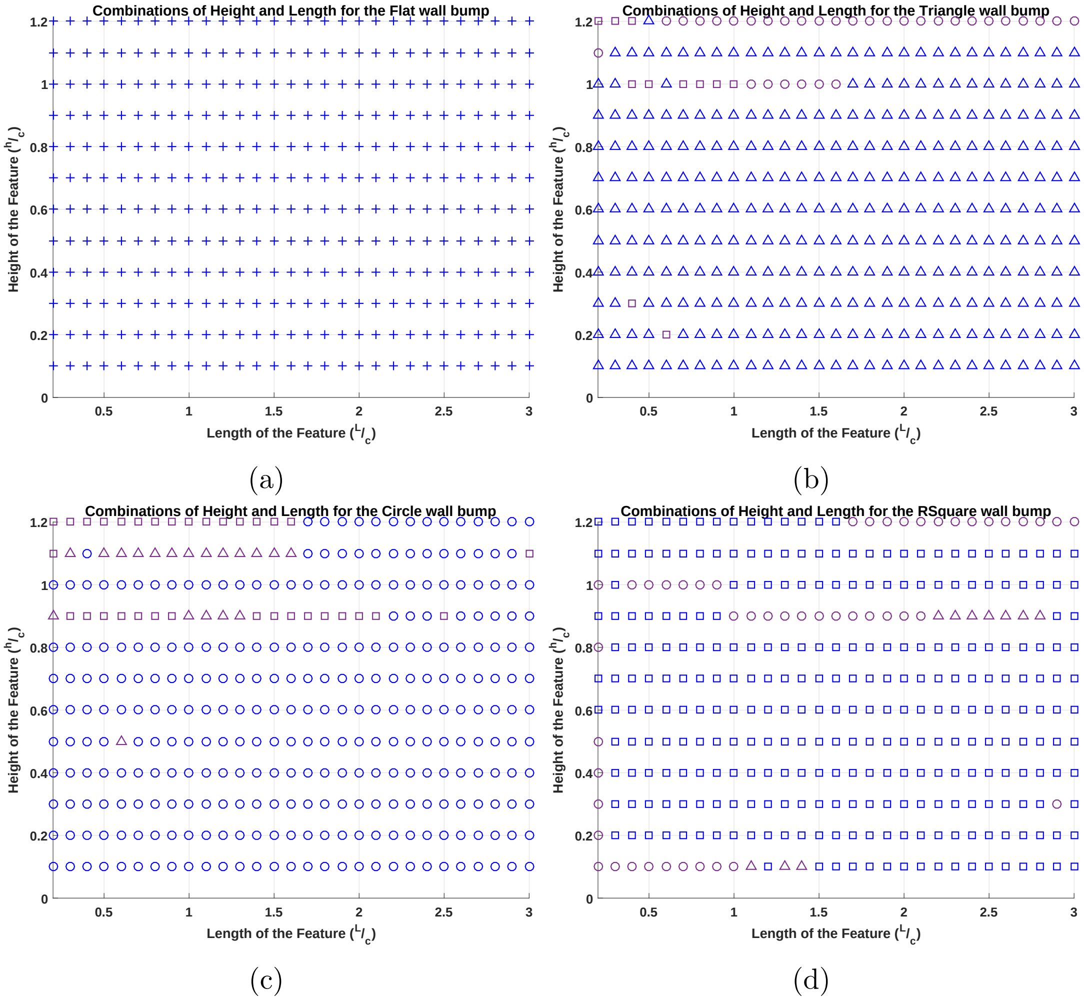


Confusion Matrix for: Support Vector Machine

True class \ Predicted class	Circle	Flat	RSquare	Triangle
Circle	**644** 23.5%		**11** 0.4%	**29** 1.1%
Flat		**625** 22.8%		**59** 2.2%
RSquare	**21** 0.8%		**632** 23.1%	**31** 1.1%
Triangle	**21** 0.8%		**1** 0.0%	**662** 24.2%

Figure 13: The confusion matrix from the classification learner. Shown is the performance of the classifier after training. Green shows correct shape classification while red shows misclassification, with percent of the total population.

A second validation step of this new classifier can be seen below in Figure 14, where all features are predicted. Blue shapes signify correct classification, while purple shapes show features that were improperly classified as a different shape. It can be seen that while the classifier does not misclassify the features as flat walls, there can be difficulty in determining the feature shape when one or both of the relevant length scales (height or length) of the feature is close to the maximum or minimum values used for training.



(a) (b) (c) (d)

Figure 14: Personal validation of the classifier using pressure signatures for all features of interest. Plus signs signify flat walls, triangles are shown with triangle markers, half-circles with circle markers, and rounded squares with square markers. A blue marker represents a correct classification, purple represents a misclassified feature as another feature, and red is a misclassified feature as a flat wall.

From Figure 13 and Figure 14, it is clear that the classifier has difficulty when a shape of a certain size looks like a different feature of a different size, e.g. a long circle looking like a rounded square. This seems reasonable when considering Figure 8, where the general shape of the pressure measurements were the same. This could lead to features of certain length scales being misclassified as features of other length scales. In order to account for the fact that size can affect the ability of the classifier to appropriately determine shape, a third classifier was developed to incorporate size distinctions.

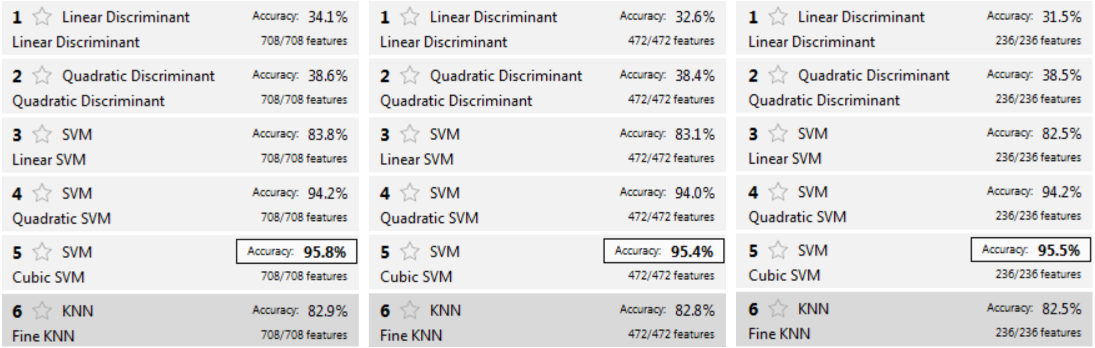
3.4 Shape and Size

To incorporate size in the classifier, a definition was chosen to signify whether shapes were small, medium, or large. Small shapes were defined to be approximately $^1/_3$ the maximum possible size or smaller, medium shapes were defined to be between $^1/_3$ and $^2/_3$ the maximum size, and large shapes were defined to be larger than $^2/_3$ the max size. The relative size of the shape could then be determined using a size assigned to both the height and the length, for example a "small" shape could have small length and height, or one small characteristic and another medium. A full explanation of all the possible size combinations can be seen below in Table 3, where "S" is small, "M" is medium, and "L" is large. Now shapes may be classified by shape and size, e.g. a triangle with small length and small height would be classified as "sTriangle".

Table 3: Combinations of height and length that create overall relative size for shapes. S means small or size $< \frac{\text{max}}{3}$, M means medium or $\frac{2\text{max}}{3} >$ size $> \frac{\text{max}}{3}$, L means large or size $> \frac{2\text{max}}{3}$.

Height \ Length	**Small**	**Medium**	**Large**
Large	M	L	L
Medium	S	M	L
Small	S	S	M

All shapes were then appended with sizes (s, m, or l) according to this schema and re-processed for training a new classifier using size and shape. This classifier was tested with the same methods as previous classifiers, in order to determine the number of pressure points necessary and which classification method worked best. It is clear from Figure 15 that the accuracy does not decrease as the number of time histories was reduced to two and further that the cubic SVM was the most accurate model.



1	Linear Discriminant	Linear Discriminant	Accuracy: 34.1%	708/708 features
2	Quadratic Discriminant	Quadratic Discriminant	Accuracy: 38.6%	708/708 features
3	SVM	Linear SVM	Accuracy: 83.8%	708/708 features
4	SVM	Quadratic SVM	Accuracy: 94.2%	708/708 features
5	SVM	Cubic SVM	Accuracy: **95.8%**	708/708 features
6	KNN	Fine KNN	Accuracy: 82.9%	708/708 features

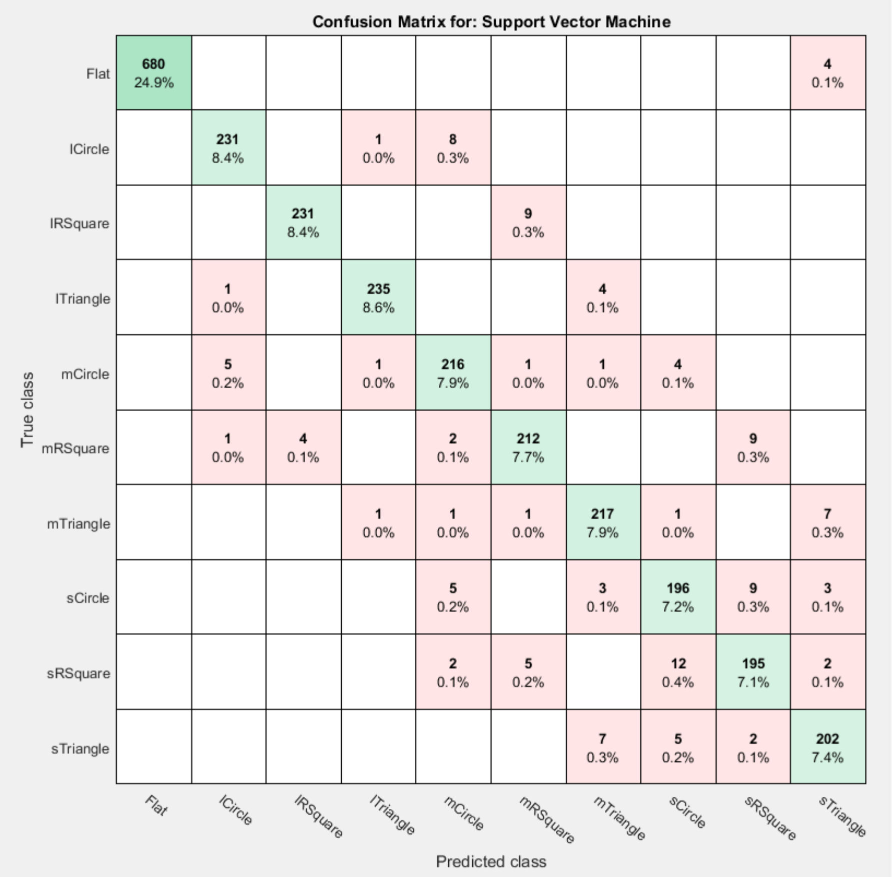
1	Linear Discriminant	Linear Discriminant	Accuracy: 32.6%	472/472 features
2	Quadratic Discriminant	Quadratic Discriminant	Accuracy: 38.4%	472/472 features
3	SVM	Linear SVM	Accuracy: 83.1%	472/472 features
4	SVM	Quadratic SVM	Accuracy: 94.0%	472/472 features
5	SVM	Cubic SVM	Accuracy: **95.4%**	472/472 features
6	KNN	Fine KNN	Accuracy: 82.8%	472/472 features

1	Linear Discriminant	Linear Discriminant	Accuracy: 31.5%	236/236 features
2	Quadratic Discriminant	Quadratic Discriminant	Accuracy: 38.5%	236/236 features
3	SVM	Linear SVM	Accuracy: 82.5%	236/236 features
4	SVM	Quadratic SVM	Accuracy: 94.2%	236/236 features
5	SVM	Cubic SVM	Accuracy: **95.5%**	236/236 features
6	KNN	Fine KNN	Accuracy: 82.5%	236/236 features

(a) All six points of pressure (b) Points 3, 4, 13, 14 (c) Points 4 and 13

Figure 15: How using different numbers of points changes the accuracy of the classifier for all wall features of interest with size.

As seen in Figure 16, a wall feature is rarely misclassified as a flat wall (0.1%) and the majority of misclassification comes from size rather than shape. It can further be seen that while the majority of predictive issues revolve around the size of the feature, there can also be issues of prediction when the shapes are very small. This is most likely due to the fact that this is at the extent of the sensing range for this type of problem, and thus pressure signatures may be less unique than if the body were closer. In order to compare the predictive ability of the network, a second validation step was performed with half of the data (Figure 17) and statistics about classification ability were computed (Table 4).



	Flat	lCircle	lRSquare	lTriangle	mCircle	mRSquare	mTriangle	sCircle	sRSquare	sTriangle
Flat	680 24.9%									4 0.1%
lCircle		231 8.4%		1 0.0%	8 0.3%					
lRSquare			231 8.4%			9 0.3%				
lTriangle		1 0.0%		235 8.6%			4 0.1%			
mCircle		5 0.2%		1 0.0%	216 7.9%	1 0.0%	1 0.0%	4 0.1%		
mRSquare		1 0.0%	4 0.1%		2 0.1%	212 7.7%			9 0.3%	
mTriangle				1 0.0%	1 0.0%	1 0.0%	217 7.9%	1 0.0%		7 0.3%
sCircle					5 0.2%		3 0.1%	196 7.2%	9 0.3%	3 0.1%
sRSquare					2 0.1%	5 0.2%		12 0.4%	195 7.1%	2 0.1%
sTriangle							7 0.3%	5 0.2%	2 0.1%	202 7.4%

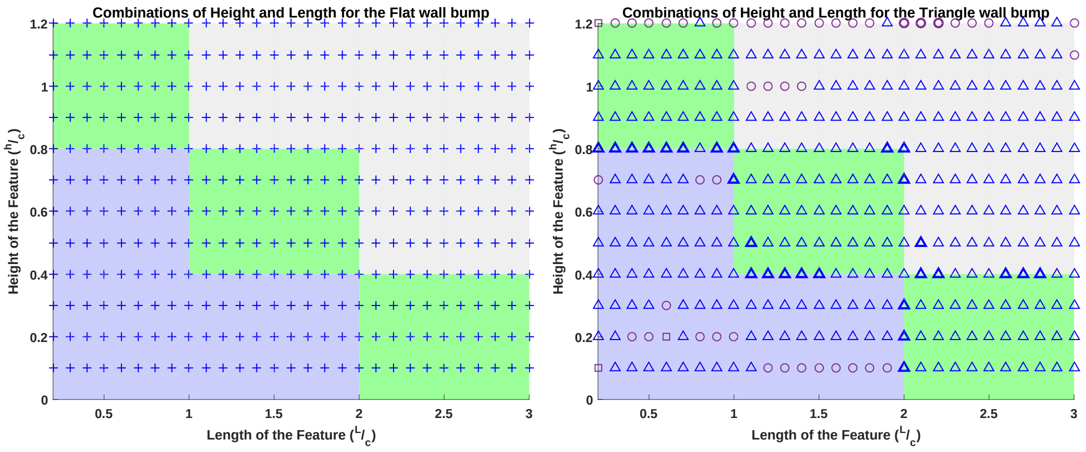
Figure 16: The confusion matrix from the classification learner in MATLAB for shape and size. The letter before each shape corresponds to its relative size.

Table 4 shows that the majority of misclassification of shape came from issues distinguishing between a circular wall and other features. This is to be expected as the shape of a circular pressure time history (Figure 8) is similar to the rounded square and triangular features, where the primary difference is that of the total magnitude of the signal. It can further be seen from Figure 17 that size was generally misclassified in regions where the shape fell on artificially imposed class boundaries, and the majority of shape misclassification occurred when the shape was near the extents (small or large) of the test matrix. The greatest errors in prediction came from attempting to distinguish between a circular and a triangular feature, especially when very small or large. While this proves to make classification of exact features difficult, it is important to note that there were almost no flat wall misclassifications and percent errors were relatively low.

Table 4: Important statistics corresponding to the second validation step. Percentages are for each discrete feature shape, while the last three statistics are for the entire population of features.

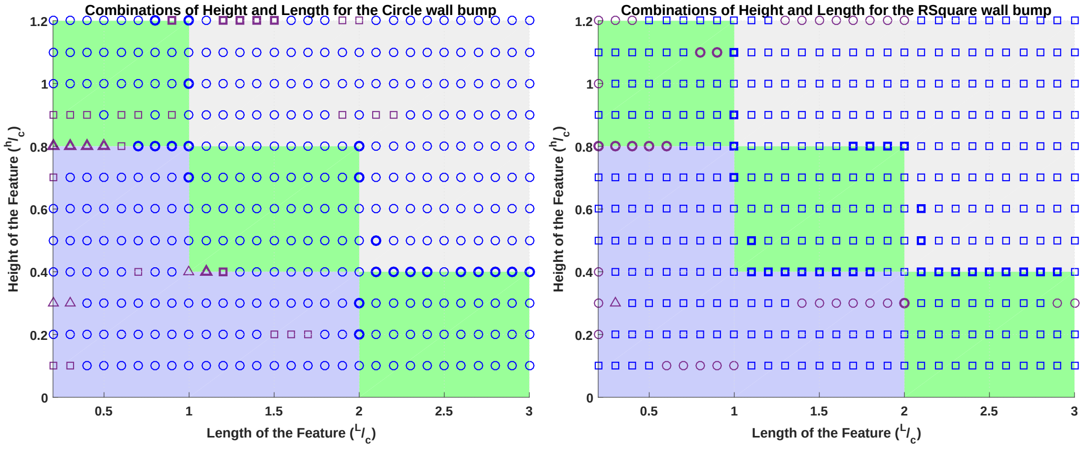
Triangle MCA RSquare	0.3%	Triangle MCA Circle	6.4%
Circle MCA Triangle	1.2%	Circle MCA RSquare	3.7%
RSquare MCA Triangle	0.3%	RSquare MCA Circle	5.5%
Wrong size: 3.4%		**Wrong shape:** 3.5%	
Both wrong: 0.7%			

Due to the fact that the majority of misclassification occurred along imposed boundaries and in regions where the shape was at the extremes of the text matrix, more complexity was added to the classifier. Previously, adding shape classification as an option made it possible for the classifier to make more refined predictions and removed ambiguity around shapes that have similar pressure signatures for distinctly different length scales e.g. a small circle being classified as a medium size triangle. This was done again in the hopes of further removing ambiguity in the solution by further refining the size.



(a)

(b)



(c) (d)

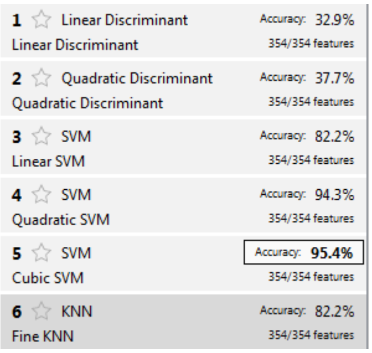
Figure 17: Validation classifier of shape and size. Plus signs signify flat walls, triangles are shown with triangle markers, half-circles with circle markers, and rounded squares with square markers. A blue marker represents a correct classification and purple shows misclassified features. The size of the marker corresponds directly to the size of the feature, with bold markers showing misclassified size. Background colors correspond to Table 3.

3.5 More General Size Classifier

For the fourth iteration of the classifier sizes were broken into 9 different categories and assigned one of three shapes such that there were 28 total unique classes, 27 shape-size combinations and one flat wall class. his allowed for an extension of the previous size regimes to include both relative length and height of the features (Table 5). Sizes were assigned to each feature by length then height, followed by a shape class e.g. a triangle with medium length and small height would be classified as "MSTriangle". Issues of misclassificaiton were still expected near imposed boundaries and close to the extents of the testing range, though overall classification was expected to improve with this added complexity. Further, it was demonstrated in Figure 10, Figure 12, and Figure 15 that anywhere between two and six pressure measurements was sufficient for this type of prediction. Due to this, measurements were used at three locations along the span of the body. It can be seen from Figure 18 that the selection of the location of pressure measurements is relatively ambiguous, as the performance of this classifier was comparable to those presented previously.

Table 5: A colored grid showing all nine possible combinations of small, medium, and large sizes. These colors will be the background on further plots.

Height\Length	Small	Medium	Large
Large	SL	ML	LL
Medium	SM	MM	LM
Small	SS	MS	LS



1 ☆ Linear Discriminant — Accuracy: 32.9%
Linear Discriminant — 354/354 features

2 ☆ Quadratic Discriminant — Accuracy: 37.7%
Quadratic Discriminant — 354/354 features

3 ☆ SVM — Accuracy: 82.2%
Linear SVM — 354/354 features

4 ☆ SVM — Accuracy: 94.3%
Quadratic SVM — 354/354 features

5 ☆ SVM — Accuracy: **95.4%**
Cubic SVM — 354/354 features

6 ☆ KNN — Accuracy: 82.2%
Fine KNN — 354/354 features

Figure 18: Accuracy of the classifier using p_4, p_7, p_{11} as specified by MATLAB.

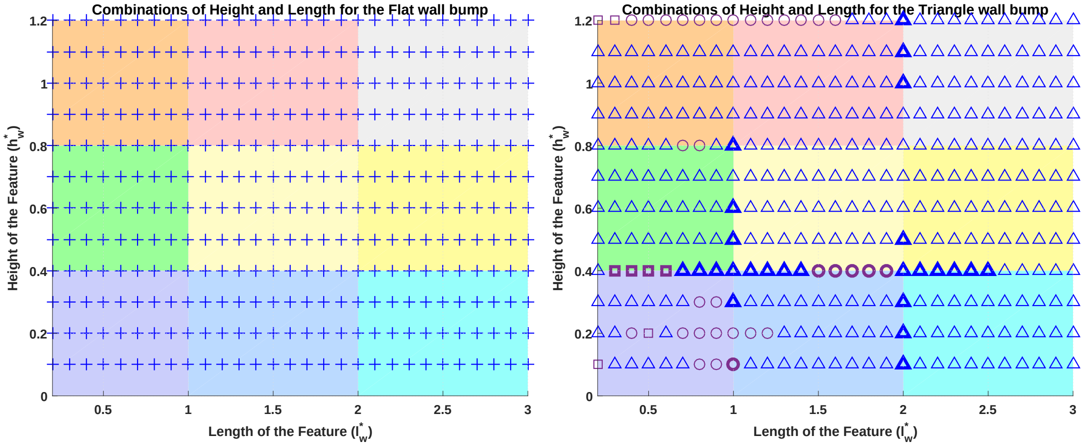
Comparing Figure 18 and Figure 12 it can be seen that the prediction accuracy of the network is unchanged by the addition of more discrete sizes. This is expected, as neural networks can make connections in data that are not otherwise obvious, and should be able to make predictions so long as it has been trained with the right information.

Interestingly when comparing Table 4 and Table 6, it can be seen that adding complexity to the classifier did not significantly change the performance. In some cases, the error of classification actually increases (Triangle MCA a rounded square) however the overall error drops. This is a common issue is the design of neural networks where added complexity can only improve predictive abilities to a certain point, after which more available information does not significantly improve classification [40].

Table 6: Statistics corresponding to the 28 class network. Percentages are for each discrete feature shape, while the last three statistics are for the entire population of features.

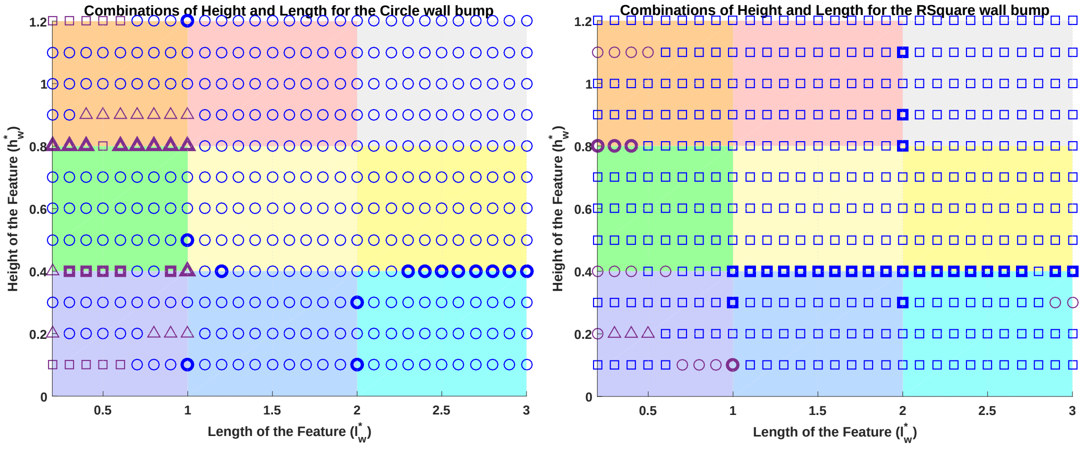
Triangle MCA RSquare	1.0%	Triangle MCA Circle	5.0%
Circle MCA Triangle	2.7%	Circle MCA RSquare	2.3%
RSquare MCA Triangle	0.4%	RSquare MCA Circle	2.0%
Wrong size: 2.7%		**Wrong shape:** 2.4%	
Both wrong: 1.1%			

Figure 19 shows again that the predominate errors in classification are found on the artificially imposed borders of the regions of classification. There are also more errors in the region of small shapes than other regions, especially for triangular features. This is again due to the similarities between the pressure time history shapes. It can also be seen that the majority of misclassifications are of size instead of actual feature shape, confirmed by Table 6. Removing some of the ambiguity in the prediction of the size of the feature slightly improved the ability to predict the shape of the feature.



(a)

(b)

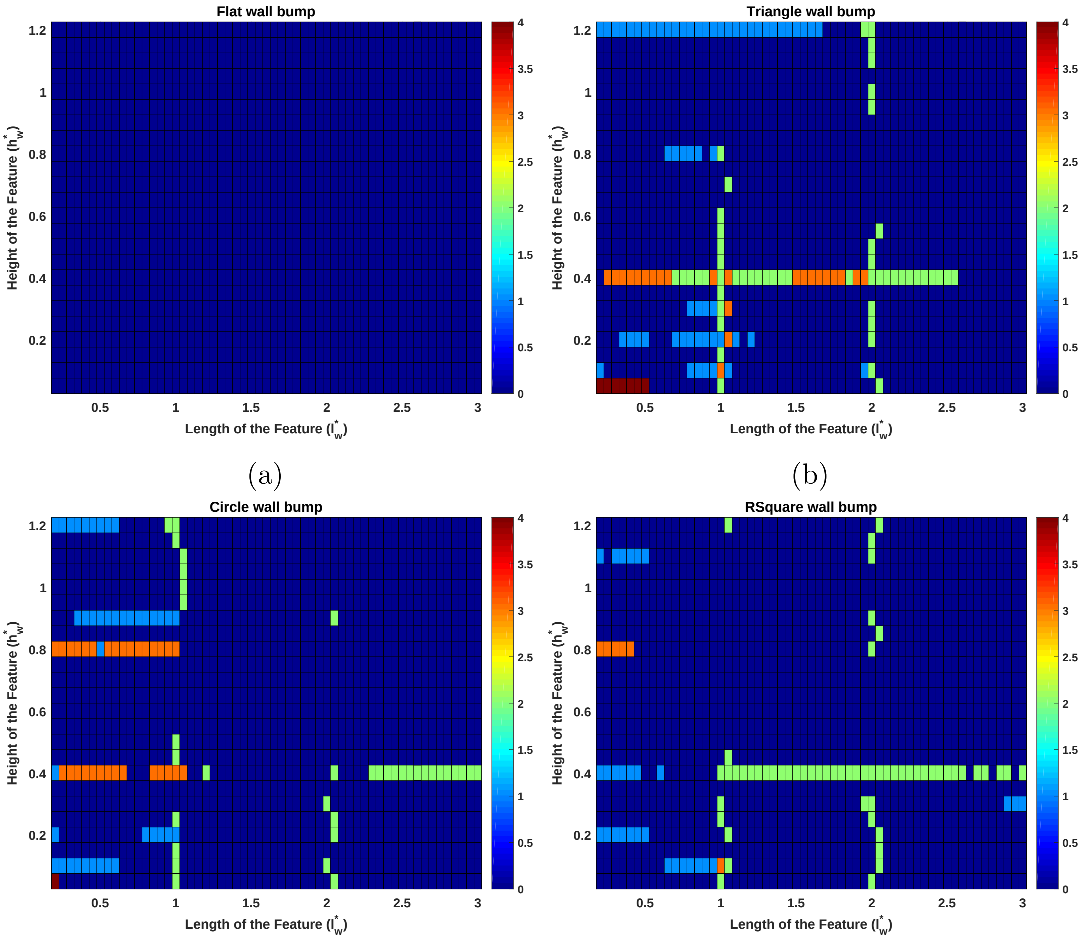


(c)

(d)

Figure 19: Validation of the classifier for 9 discrete sizes and 3 shapes. Plus signs signify flat walls, triangles are shown with triangle markers, half-circles with circle markers, and rounded squares with square markers. A blue marker represents a correct classification, purple represents a misclassified feature as another feature, and red is a misclassified feature as a flat wall. The size of the marker corresponds directly to the size of the feature, with bold markers showing misclassified size. The background color corresponds to Table 5.

Due to the discretization of the classes, the classification learner confusion matrix is difficult to understand. For this reason, a similar analysis was performed (Figure 20) in order to observe the error space of the classifier. Figure 20 clearly demonstrates that misclassified sizes occur almost entirely on imposed size boundaries, and misclassified shapes occur predominantly in regions where either length or height is small. It is obvious why there are issues of classification for size on imposed boundaries because features that lie on this boundary could easily be classified as the size on the other side of the boundary. Less obvious was that when shapes were in the small-small region, or either length or height was small, features appear similar in shape and thus pressure signatures are less unique. Further, if the shape is very small, it can be misclassified as a flat wall as seen in Figure 20b and Figure 20c. This is due to the shape being at the extent of the sensible range for this body and thus indistinguishable from a flat wall. When the shape is sufficiently large (as in the upper right corner of the subplots in Figure 20) it is easy to make an estimate of shape with almost no misclassification of the feature. This indicates that allowing the object to be close to the foil helps with identification. Allowing the foil to move vertically towards the wall could therefore help with classification but would require additional classes for distance from the wall.



(a)

(b)

(c)

(d)

Figure 20: Error space for the 28 class network. 0 is correct classification. 1 is incorrect shape. 2 is incorrect size. 3 is incorrect shape and size. 4 is flat wall misclassification.

3.6 Discussion

From the progression of the classifier, it can be seen that pressure is capable of measuring object information. Further, it appears that as the number of available classes in the training set increased, the classifier improved. The majority of the prediction errors were observed to be from misclassifying the size of the object. This implies that if the exact size (e.g. the height and length in units rather than a relative size scale) and shape of each feature was used to classify the shapes, than the network would perform better overall.

While it is clear that this method works for object recognition of this type, it would be increasingly difficult to train a network to make general types of classification with this method. A classifier that could make recognitions in this manner would require a nearly infinite number of different class options including characteristics that have not been considered in this investigation. It is also difficult to validate the results of a neural network as the internal workings of the system are difficult to interpret and thus a transparent solution is often impossible to provide. While the neural network classifier indicates that pressure time histories are an effective proxy for identifying wall features, the method is not suitable for general characterization of unknown wall features. A more general method for estimating wall features is developed in the next chapter.

CHAPTER 4

B-spline UKF Estimator

In this chapter, the general B-spline representation of a shape is used with an Unscented Kalman filter (UKF) in order to estimate a general wall shape over time. For the classifier, a set of predefined wall protrusions were simulated in LilyPad and the pressure measured on the surface of the foil was used to infer class. Rather than design a wall with predefined shapes, the wall is described with a B-spline curve. B-splines, calculated using Equation 1, are 2-D line segments which are described by a set of control points multiplied by a set of weighted basis functions [41].

$$S_{m,\mathbf{t}}(x) = \sum_{i=0}^{D-1} P_i B_{i,m}(x) \tag{1}$$

Where $S_{m,t}(x)$ is a particular spline segment of polynomial order m at point x relative to the knot vector $\mathbf{t}$, P_i are the D control points, and $B_{i,m}(x)$ is the basis function calculated relative to the knot vector (Equation 2 and 3).

$$B_{i,0}(x) = \begin{cases} 1 & \text{if } t_i \leq x \leq t_{i+1} \text{ and } t_i \leq t_{i+1} \\ \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$B_{i,j}(x) = \frac{x - t_i}{t_{i+j} - t_i} B_{i,j-1}(x) + \frac{t_{i+j+1} - x}{t_{i+j+1} - t_{i+1}} B_{i+1,j-1}(x) \tag{3}$$

The wall in the simulation is now defined by a set number of D control points which locally govern the shape of the solid boundary in LilyPad. Representing the wall with B-splines allows general shapes to be made by modification of control point locations, and provides a natural extension to three-dimensional NURBS surfaces for 3-D problems. For simplicity, the number of knots can be calculated based on the order of the polynomial and the number of desired control points such that $\mathbf{t}$ contains $D + m + 1$ knots, or weights.

4.1 The Unscented Kalman Filter

The *UKF* is an extension of the Linear Kalman Filter, which has the ability to account for highly non-linear state and measurement relationships by assuming Gaussian statistics for the unknown states [42]. The UKF uses a sigma transform and an unscented transform to estimate the first and second moments of the random variable measurement and state distributions, while limiting the number of random realizations necessary to obtain a statistical representation of these variables. The sigma transform and unscented transform are able to account for high order statistical moments with low added computational cost.

For this method, it is assumed that the locations of control points that define a B-spline wall in a simulation are the unknown system state variables that we wish to estimate. Measurements of pressure on the surface of the foil body moving through the simulation are used as observations at known instances of time. In order to make an estimate of the control points, a state equation (Equation 4) and a measurement equation (Equation 5) are assumed:

$$\mathbf{P}[k+1] = \mathbf{P}[k] + \mathbf{v}[k] \tag{4}$$

$$\mathbf{p}[k] = F(\mathbf{P}[k]) + \mathbf{w}[k] \tag{5}$$

where k is discrete time, $\mathbf{P}$ are the vertical position of N control points with distribution covariance C^P, $\mathbf{p}$ is a time history of pressure modeled at defined locations on a foil that are the output of the non-linear process F. In this case F is the numerical CFD simulation (LilyPad) that relates the location of control points $\mathbf{P}$ to the measured pressure $\mathbf{p}$. The covariance of the current time history of pressure at a location on the foil is given by C^F, and $\mathbf{v}[k]$ and $\mathbf{w}[k]$ are uncorrelated additive white noise on the state and measurement respectively [42]. The state and measurement noise have variances q and r, and covariance of the form $Q = qI$ and $R = rI$ where Q is size $N \times N$ and R has size on the order of the size of the measurement, or $\mathcal{O}(\mathbf{p}^F)$.

In order to simplify the problem for development of the algorithm, the control point locations were placed at evenly spaced, known horizontal positions, such that only the vertical position of each control point was unknown. To satisfy the assumption of Gaussian statistics, a time history of $M \geq 50N$ measurements per time history was used in the update stage of the *UKF*.

4.1.1 The Sigma Transformation

The sigma transformation is a method of selecting a set of statistically relevant weighted sample points to capture the statistics of a random variable distribution [42]. These points contain the necessary information about the true mean and covariance of the random variable by design of the transform. The number of sigma points (L) are determined through the number of states by the relationship $L = 2N + 1$, which allows for the addition of more estimated states with little additional computation needed. Wan and Van Der Merwe [42] show that each sigma point can be calculated based on the estimated mean state, $\bar{\mathbf{P}}$, and the estimated state covariance C^P:

$$X_0 = \bar{\mathbf{P}} \tag{6}$$

$$X_i = \bar{\mathbf{P}} + (\sqrt{(N+\lambda)C^P})_i, \ i = 1, ..., N \tag{7}$$

$$X_i = \bar{\mathbf{P}} - (\sqrt{(N+\lambda)C^P})_{n-i}, \ i = N+1, ..., 2N \tag{8}$$

where $\lambda = \alpha^2(N + \kappa) - N$ is a scaling parameter corresponding to the mean and covariance weights, W^m and W^c in Equation 9 and Equation 10. The unscented transform constant $\alpha > 0$ controls the spread of the sigma points about the estimated mean state and is typically a small number, though it can vary depending on the characteristics of the estimated state variables. A value of $\kappa \geq 0$ guarantees that the estimate of the covariance will remain positive semi-definite, but can vary depending on the state variables.

$$W_0^m = \frac{\lambda}{L+\lambda},\ W_i^m = \frac{1}{2(N+\lambda)},\ \text{for } i = 1, \ldots, 2N \tag{9}$$

$$W_0^c = W_0^m + (1 - \alpha^2 + \beta),\ W_i^c = W_i^m \tag{10}$$

$\beta \geq 0$ is used to statistically incorporate information of higher order moments of the distributions, and $\beta = 2$ is optimal for the standard normal distribution. Wan and Van Der Merwe [42] provide suggested values of α, κ, and β but it is important to note that they must be tuned for specific problems.

4.1.2 The Unscented Transform

The unscented transform is a method to estimate the mean and covariance of a transformed random variable distribution from the sigma points using the defined weights. Using the sigma points as determined above, the sigma points are passed through the state and measurement equations, Equation 4 and Equation 5. This allows for the calculation of μ_P and μ_F, the current estimate of the state and measurement random variables, as a weighted average based on the sigma points as in Equation 11 and Equation 12.

$$\mu_P = \sum_{i=0}^{2N} W_i^m X_i \tag{11}$$

$$\mu_F = \sum_{i=0}^{2N} W_i^m p_i^F \tag{12}$$

where p_i^F represents the pressure time history from a numerical simulation run using the boundary conditions, X_i, determined through the sigma transformation. After calculating the mean of each random variable, it is possible to determine the covariance matrices corresponding to each distribution as in Equation 13 and

Equation 14.

$$C^x = \sum_{i=0}^{2N} W_i^c (X_i - \mu_P)(X_i - \mu_P)^T \tag{13}$$

$$C^y = \sum_{i=0}^{2N} W_i^c (p_i^F - \mu_F)(p_i^F - \mu_F)^T \tag{14}$$

The unscented transform is designed such that it passes the first and second moment of the statistical distributions of the random variables that represent the system through the non-linear relationships between state and measurement [42]. This process captures the mean and covariance of the system random variables as determined by the sigma points, X_i. When combined with the Kalman Filter, the estimates from the unscented transform can be corrected based on the error between the estimated measurement and the observed measurement. Combining the sigma and unscented transforms provides accurate estimates with fewer random variable realizations than would be necessary for statistically sampling a random variable. For example, the transform allows for only a handful of realizations to be performed to capture the random variable statistics rather than hundreds or thousands. This is important when running numerical simulations that require significant processing efforts.

From Wan and Van Der Merwe [42], the cross-variance between the state and measurement estimates must be calculated (Equation 15) to make a correction to the estimates from the unscented transform.

$$C^{x,y} = \sum_{i=0}^{2N} W_i^c (X_i - \mu_P)(p_i^F - \mu_F)^T \tag{15}$$

The gain of the filter, K, is then calculated from the cross-variance as in Equation 16 and used to correct the mean of the state (Equation 17) based on the measurement

$\mathbf{p}^{meas}$ and the covariance (Equation 18).

$$K = C^{x,y}(C^y)^{-1} \tag{16}$$

$$\bar{\mathbf{P}} = \mu_P + K(\mathbf{p}^{meas} - \mu_F) \tag{17}$$

$$C^P = C^x - K(C^{x,y})^T \tag{18}$$

4.2 B-Spline Estimation Algorithm

In order to implement the *UKF*, a set of a priori decisions were made about the definitions of the state and measurement for the start of estimation. In the B-spline definition (Equation 1) an arbitrary number of control points, D, can be defined to describe the surface of the wall. This allows for any amount of discretization of the wall, though the number of states to be estimated increases proportionally to the number of control points. This poses a problem as increasing the number of states increases the number of sigma points, and thus the number of numerical simulations that need to be run.

A useful feature of the B-spline curve definition is that the position of points on the curve are only determined by the position of local control points such that wall shape is only locally defined by $m+1$ control points [41]. This implies that the curve does not need to be globally estimated, but could instead be locally estimated based on a subset of N control points near the foil body that shift in location as the body propagates forward in time.

To initialize the method, the wall was assumed to be flat such that the D control points which describe the surface of the wall are set to a vertical position of zero and assumed an identity matrix for the state distribution covariance matrix. The horizontal position of each control point is then defined by the discrete time step, k, such that the horizontal location of each control point is defined by Uk, where U was the forward speed of the foil. It should be noted that the discrete time k does not correspond to simulation time, and instead corresponds to the global time which defines the current position of the foil for the observed signal.

At every time step, k, the sigma points for the current window of N control points are calculated in order to define the L simulations which are used to compute the statistics of the estimated random variables. The sigma point transformation provides L sets of control point locations which are used to define the wall shape in simulations. In this way, the statistics associated with the current set of N control points are calculated based on the variability in the pressure measured in the L simulations. The L simulated time histories of pressure were then used to update the vertical positions of the N control points using the previously stated gain correction procedure (Equations 15 - 18).

Once an update to the current window of control points is made, the root mean square error (RMSE) is calculated between the measured value of pressure and the estimate from the *UKF*, μ_F. This process is repeated using the current window of control points until a convergence criteria has been met based on the calculated RMSE. In this study, the convergence cutoff was defined as the iteration at which the percent error between the current iteration and the previous iteration changes by less than one percent. This is synonymous with the criteria that the current estimate of the states of the system are no longer being substantially modified using the current set of measurements. Here, n control points on the downstream side of the window are removed from the window and no longer allowed to change. The foil then moves forward in time and n new control points on the upstream side of the estimate are added to the window such that N control points are again being estimated (Algorithm 1 and Figure 21). For further explanation of the variables used in the formulation of the *UKF*, see Table A.1 in Appendix A.

An important aspect of fluids is that they contain memory, such that the pressure at a single time on the foil is dependent on the time history of the wake. This means that while the true foil is moving forward in time, every simulation run in the unscented transform algorithm needs to be initialized from the same time in order to maintain the history of the wake. It is also necessary to keep a history of previously estimated states in order to maintain the wall shape downstream of the estimate window in each simulation. This causes the simulation time to increase as the number of sigma parameters being estimated increased. Once control points become locked in position, i.e. the window of N control points has moved so a local point is no longer allowed to move, the shape of the wall becomes fixed in that location. To reduce computational time, the simulations could be initialized to a start position later in time based on the fixed wall shape, however this is not implemented in the current version of the algorithm and will be an aspect of future work.

4.3 Importance of Spline Order

Due to the design of the method, it is necessary to keep the order, m, of the spline curve estimate low. This is due to the local definition of the spline curve based on control points, where any local point on a curve is calculated based on the closest $m+1$ control points. This means that spline curve points that lie on the end points of the estimation window are controlled by control points which have either been fixed on the downstream side of the window or have yet to be estimated on the upstream side. Increasing the order of the polynomial will increase the number of these control points that lie outside the window, and thus it is desirable to keep the spline order low (e.g. 3 or 4). Keeping this value low maintains the ability of the prediction to be done in a window region as it minimizes the affect that control points outside the window have on the current set of control points allowed to move.

Algorithm 1 B-Spline UKF Estimator

```
1:  procedure Initialization
    return $N, n, Q, R, \lambda, W^m, W^c, \bar{\mathbf{P}}, C^P, \mathbf{p}^{meas}$
2:  function Sigma($\bar{\mathbf{P}}, N, \lambda, C^P$)
    return $X_i$, for $i = 0, \ldots, L$
3:  function Unscented($W^{m,c}$, $\mathbf{X}$,Eq 4)
    return $\mu_P, C^x$
4:  procedure Run Sigma Simulations
5:      Run simulation for each $X_i$ concurrently
    (Include previously estimated control points)
    return $p_i^F$ for each $X_i$
6:  function Unscented($W^{m,c}$, $\mathbf{p}^F$,Eq 5)
    return $\mu_F, C^y$
7:  procedure Gain Update($W^c, \mathbf{X}, \mu_P, C^x, \mathbf{p}^F, \mu_F, C^y$)
    return $\bar{\mathbf{P}}, C^P$
8:  procedure Compare Error
9:      Calculate $RMSE(\mathbf{p}^{meas} - \mu_F)$
10:     Calculate the percent different in RMSE
11:     if Percent difference $\leq$ threshold then
12:         Save $P_i \rightarrow P_n$ control points
13:         Shift $P_{n+1}$ to $P_1$, add $n$ control points
14:         Shift the covariance matrix
15:         Step $k$ forward in time
16:         Re-sample measurement
17:     Return to step 2
```

4.4 Important Characteristics

In order to assess the performance of the estimation, a variety of error metrics are observed in the flat wall regions of the boundary estimates and in regions where features exist. Since this method is developed in the interest of digital wins as well as object detection, the performance is considered from both the point of object recognition as well as boundary identification. For this reason, error metrics are defined based on the estimation of the width, height, and center location of the feature. Further, it was of interest to determine the minimum distance between the foil body and boundary estimate as well as the total RMSE between the full shape estimate and the true wall shape.

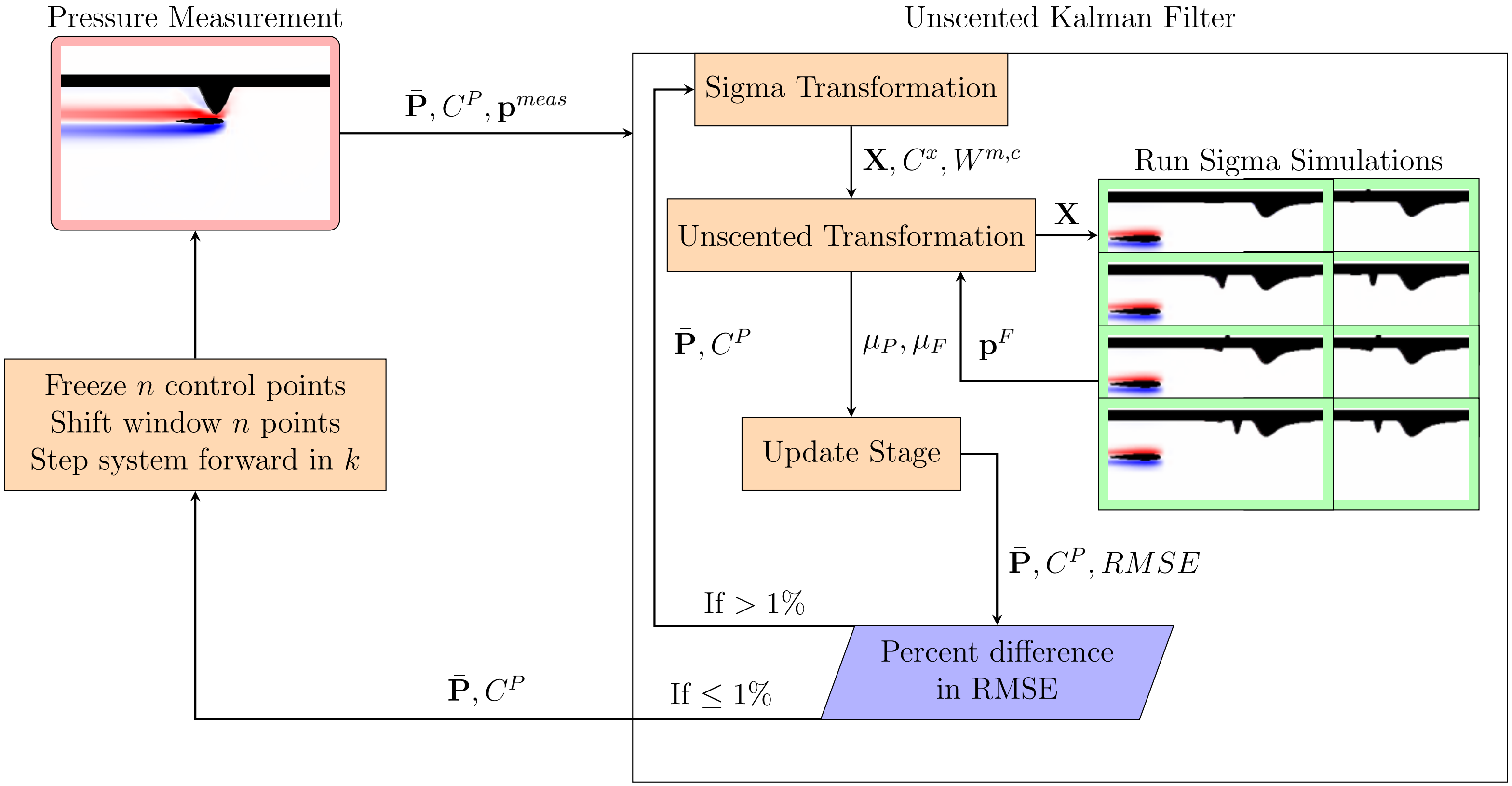


Figure 21: Flow chart illustrating the B-spline *UKF* method.

CHAPTER 5

General Object Estimation Using UKF Algorithm

5.1 One Feature

We visit again the problem of a streamlined body passing by a protrusion from a flat wall. A general shape was defined with $D = 124$ control points (Figure 22), and ten control points in a local area modified to make a smooth, continuous protrusion from the wall. The protrusion was defined such that the total length of the feature, l_w^*, was less than one chord length, and the height, h_w^*, was chosen such that the body could be placed at a foil wall distance of $H^* = 0.75$ without collision. This guaranteed that the flat wall regions were within the sensing range defined by Bouffanais et al. [32] for the entirety of each simulation. The important characteristics that define the feature are shown in Table 7 where H is the height of the feature, W is the width, dis is the minimum distance between the wall and the foil body, and cen is the location of the center of the feature in non-dimensional units (chord lengths).

Table 7: Characteristics of the feature to be estimated. Height and width are in units of chord lengths, distance and center are in terms of the physical location of the foil referenced from the start position.

	H	W	dis	Cen
Actual	0.59	0.48	0.16	102.5

A window for $N = 19$ control points was chosen, with the window shifting by $n = 1$ control points for each discrete time step. Based on the work presented by Harashima et al. [43] and Jauch et al. [44] in recursive spline estimation using Kalman Filters, it was decided that the noise level on the state would be set as $Q = \underline{0}$ and the noise level for the measurement was set proportional to the current LilyPad resolution, such that $R = 0.5 \times 10^{-3} I$. Further, following Wan and Van

Der Merwe [42] and Julier [45], for this investigation it was decided to set the *UKF* constants to $\alpha = 1,\ \beta = 5,$ and $\kappa = 1$. The effect of these constants on the solution are evaluated later.

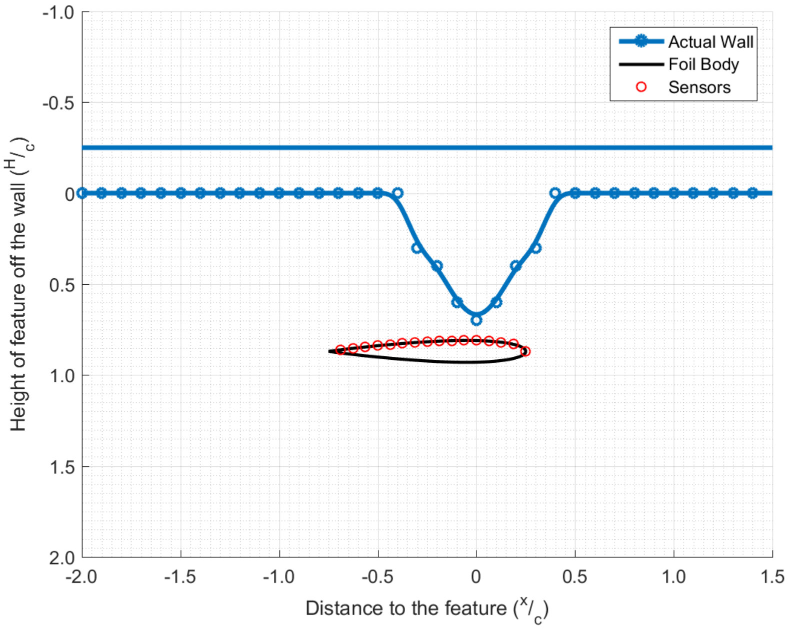


Figure 22: The wall shape to be estimated using the B-spline UKF algorithm. Pressure is measured at the shown locations along the foil body.

5.1.1 Standard b-Spline UKF

Initially, to estimate the B-spline we use a standard formulation of the *UKF*, where each pressure measurement at some given point in time is considered to be an observation of the system. Using this method the measurement array is constructed as a single column array, $\mathbf{p}^{meas}$:

$$\mathbf{p}^{meas} = \begin{bmatrix} \mathbf{p}_1 \\ . \\ . \\ . \\ \mathbf{p}_{16} \end{bmatrix} \tag{19}$$

where each $\mathbf{p}_i,\ i = 1, \ldots, 16$, is a column array corresponding to the time history at the i^{th} measurement location of the foil. This method uses all pressure measurements over the M length time window to estimate the current states (vertical position of control points) and applies equal weight to each location of pressure along the length of the body.

Figure 23 shows the estimate of the wall at three discrete time steps k (three separate locations of the foil) in three colors, based on the presented algorithm and standard measurement design. The colored shapes near the lines correspond to the locations of the control points in the estimation window for that time step.

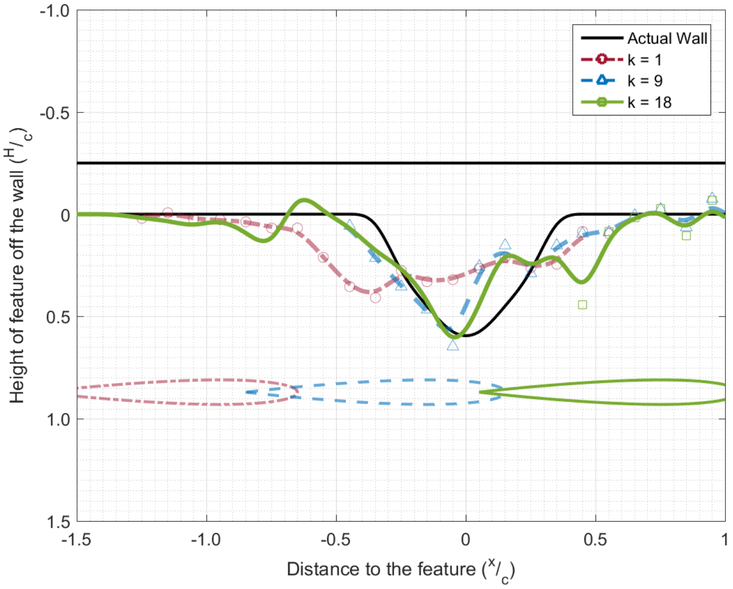


Figure 23: The standard *UKF*. Estimates of wall shape with foil position are shown at 3 discrete time steps of the filter, where color denotes each time step and shapes (circles, triangles, and squares) show control point locations at that time step.

Figure 23 shows that the method is able to identify an object on the wall with rough characteristics (height, width, location) of the true shape, though the method is poor at estimating true shape. The height of the protrusion is estimated best but the width has some error, though the overall size and location are estimated well based on the comparison of the shape at $k = 18$ and the actual location of the shape. There are significant fluctuations in the upstream and downstream estimates where the true wall is flat and large deviations are observed on the back side of the feature.

The fluctuations on the flat wall are partly due to the fact that the sensing body is relatively far from the flat wall, where sensitivity to differences in pressure will be small as discussed by Bouffanais et al. [32], however this does not explain the large error in shape observed on the back of the feature. Instead, since our estimate is based on a proxy of the wall shape through pressure measurements, errors in the wall shape that don't significantly alter the pressure (such as fluctuation in the wall where it should be flat) don't significantly affect the estimate. Pressure measurements in close vicinity to the true signal are then more valuable, thus the algorithm needs to be altered to give more weight to pressure measurements with more information. To do this, the formation of the measurement array was altered such that the estimate of the wall shape was derived from each location of pressure on the foil body individually, where each estimate could be weighted based on its ability to match the pressure signal.

5.1.2 Weighted b-spline UKF

Instead of calculating a single estimate of the control points in the current window based on all measurement locations, each time history of pressure $\mathbf{p}_i$ was used to calculate an individual estimate of the wall shape. This was done by modifying the update phase of the *UKF* such that there were 16 individual estimates of the control point locations in the current window, each corresponding to a specific pressure time history measured on the body of the foil. The estimate of the wall shape was then calculated based on a weighted average of all the wall estimates from each pressure time history. In order to calculate the weights, W^e, the RMSE between the observed and estimated pressure at each point i on the foil was calculated. The RMSE was then inverted and normalized to give W_i^e, a weight that lies between 0 and 1, with the sum of the weights over the i pressure measurements equal to 1. This is implemented as an expansion to the gain update formulation and pressure measurements with low RMSE to be given more preference in determining the estimated shape. This method is detailed in Algorithm 2.

Algorithm 2 Weighted Gain Update

```
procedure GAIN UPDATE
    for $j = 1, ..., 16$ do
        $C_j^{x,y} = \sum_{i=0}^{2N} W_i^c (X_i - \mu_P)(p_{i,j}^F - \mu_{F,j})^T$
        $K_j = C_j^{x,y} (C^y)_j^{-1}$
        $\bar{P}_j = \mu_P + K_j(\mathbf{p}_j^{meas} - \mu_{F,j})$
        $C_j^P = C^x - K_j(C_j^{x,y})^T$
        $w_j^e = 1/[\sqrt{\frac{1}{M}\sum(\mathbf{p}_j^{meas} - \mu_{F,j})^2}]$
    $W^e = \frac{\mathbf{w}^e}{\sum_{j=1}^{16} w_j^e}$
    $\bar{\mathbf{P}} = \sum_{j=1}^{16} W_j^e \bar{P}_j$
    $C^P = \sum_{j=1}^{16} W_j^e C_j^P$
return $\bar{\mathbf{P}}$, $C^P$
```

As seen in Figure 24, there is a marked improvement in the estimation of both the size and location of the feature. As opposed to Figure 23, one can see that the height and width of the feature is captured well, and fluctuations in the flat wall regions are on a smaller scale, though they have not been completely eliminated. It should be noted that the weighted average method results in a slower change in the overall shape estimate over time, which can help stabilize the estimate. While the overall estimate with the weighted *UKF* shows improvements over the initial method, there are still undesirable fluctuations in the flat wall regions and a slight dip in the peak of the solution.

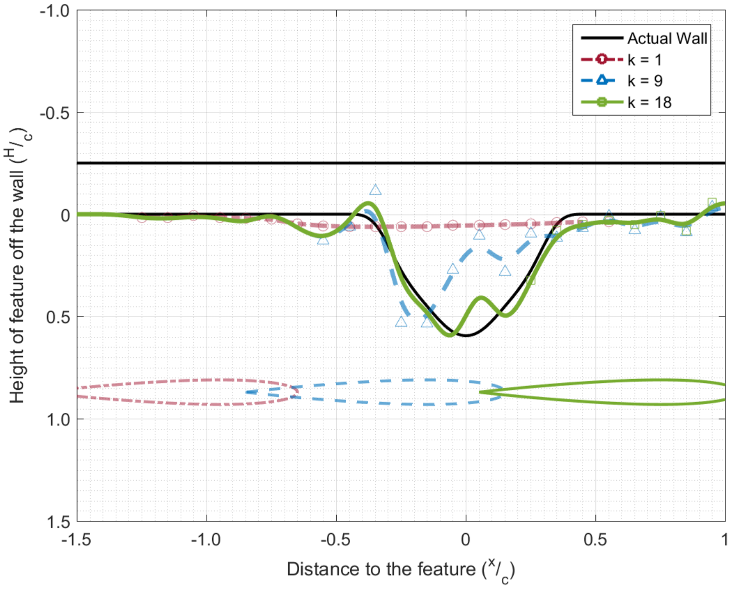


Figure 24: The weighted *UKF*. Estimates of wall shape with foil position are shown at 3 discrete time steps of the filter, where color denotes each time step and shapes (circles, triangles, and squares) show control point locations at that time step.

Weighting each estimate individually improved the estimate of the shape, width, and height, however other factors still influence error in the solution. In particular, with a B-spline representation of shape, even with only one control point out of place the shape can have significant error. This could easily be affected by local minima in the solution and choice of initial conditions. Due to the ability of the weighted average method to nearly capture the true shape of the feature, further refinement of the method and implementation in other examples is based on the weighted average technique.

5.1.3 Performance Analysis

Previously, error in predictions was discussed based on how the estimates visually compared to one another, however this does not provide a measurable performance of the algorithm. Following a similar analysis as Maertens [46], Table 8 provides a quantifiable error analysis for the standard and weighted measurement methods. The following error metrics are presented in non-dimensional units of chord lengths and percent error:

1. H - The height of the estimated feature off the flat wall.

2. W - The width of the estimated feature.

3. dis - The minimum distance between the foil body and the estimated feature over the full time history of the estimate.

4. Cen - The estimated location of the center of the feature referenced from the simulation start point.

5. $\tilde{H}$ - The percent error of the height compared to the true shape.

6. $\tilde{W}$ - The percent error of the width compared to the true shape.

7. $\tilde{dis}$ - The percent error of the minimum distance between the foil and feature compared to the known distance.

8. $\tilde{Cen}$ - The percent error of the center of the feature compared to the true shape.

9. RMSE$_T$ - The total RMSE of the spline estimate, including the flat wall.

10. RMSE$_F$ - The RMSE of the estimate only in the region close to the feature.

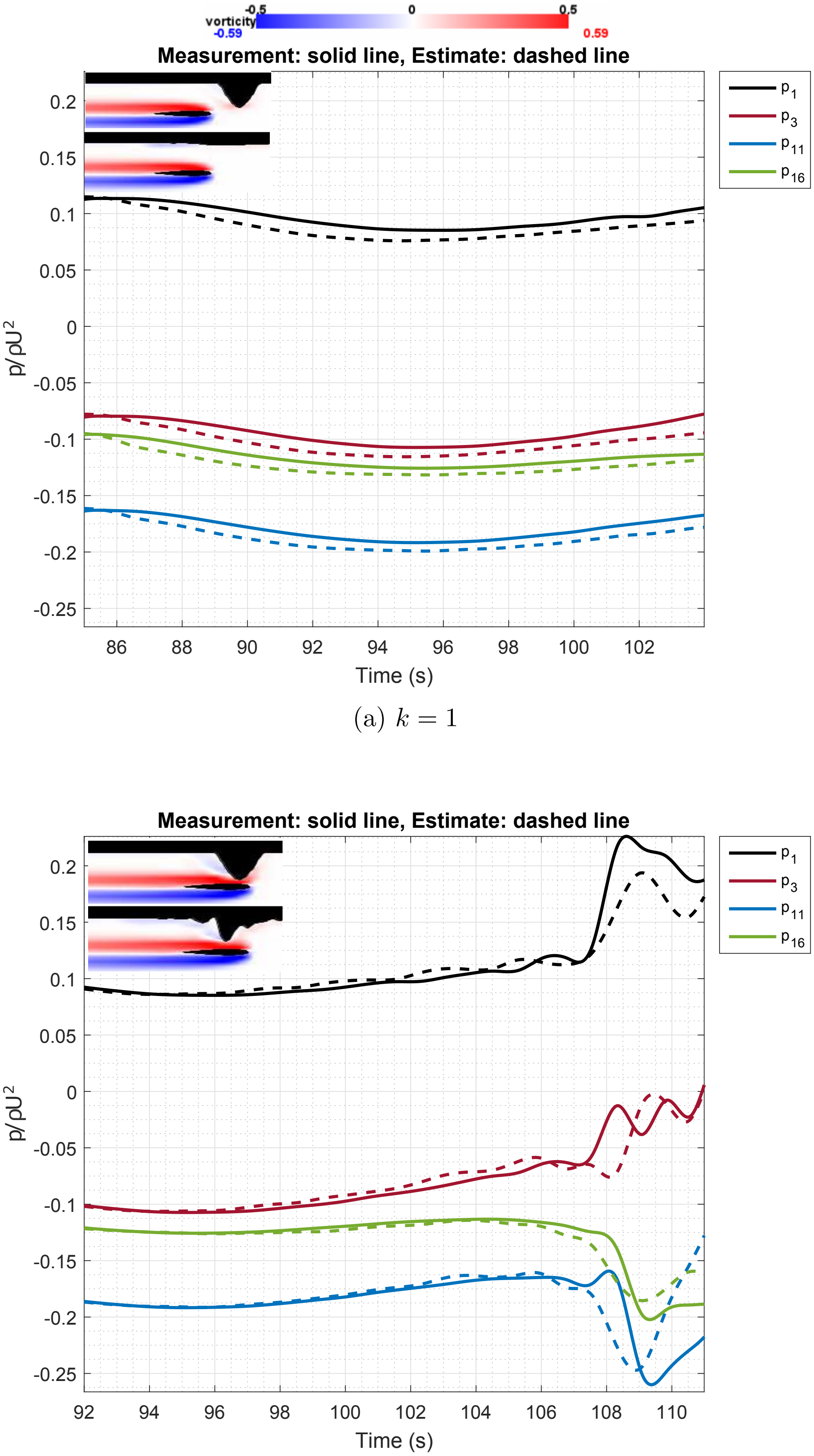
These ten metrics give a good measure of the performance of the method by identifying quantifiable errors about general features, as well as the quality of the overall spline estimate. From Table 8, one can see that the height of the feature, the location of the center of the feature, and the minimum distance between the foil and the feature was well captured for both the standard and the weighted measurement method, which was anticipated based on visual inspection. However, the standard measurement method shows a 44% error in estimation of the width of the feature, much higher than the 12% error in width using the weighted method. Overall, every general error metric is reduced when the weighted average implementation. For the overall spline estimate, the weighted average method also reduces the local feature RMSE$_F$ by a factor of two and reduces the total RMSE$_T$ by a factor of three, demonstrating the improvement in estimation using the weighted method.

Table 8: Error metrics for the standard and weighted *UKF* methods. Height, width, and distance are in units of chord lengths, and the center is defined by the physical location of the center referenced from the start position of the simulations. The characteristics of the true wall are also presented for comparison.

Method	H	W	dis	Cen	$\tilde{H}$	$\tilde{W}$	$\tilde{dis}$	$\tilde{Cen}$	RMSE$_F$	RMSE$_T$
Actual	0.59	0.48	0.16	102.5	-	-	-	-	-	-
Standard	0.60	0.69	0.15	103.9	0.013	0.44	0.05	0.014	0.126	0.063
Weighted	0.59	0.53	0.16	103.0	0.003	0.12	0.01	0.005	0.076	0.026

5.2 Pressure as a Measurement

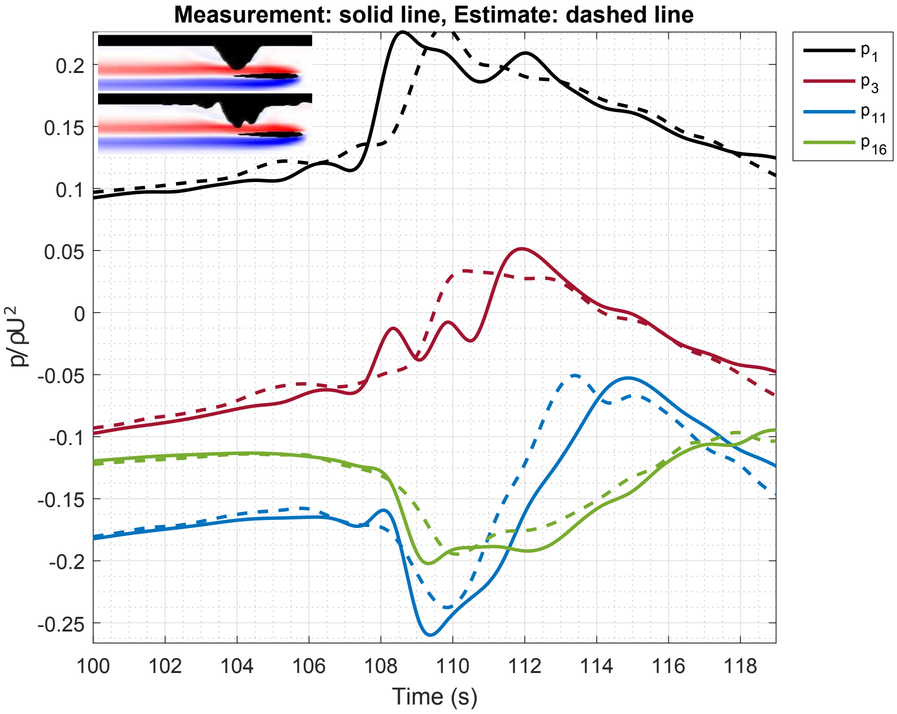
Although estimating wall shape is the ultimate goal in this estimation method, wall shape is estimated through the proxy measurement of pressure. For an exact determination of the location of each control point, the pressure signature estimated from the unscented transform would need to exactly match the observed signal for all sensor locations on the body for all points in time, an unlikely scenario in any Bayesian estimation method. Figure 25 shows four of the pressure time histories used in the weighted *UKF* method at the same three values of the discrete time step k from Figure 24. The four time histories of pressure are shown in different colors, with a solid line representing the measurement signal and a dashed line showing the estimated pressure (μ_F). A LilyPad screen capture of the actual (top) and estimated (bottom) wall shape at timestep k is shown in the upper left corner of each plot with vorticity described by the color bar above the first plot.



vorticity
-0.5
0
0.5
-0.59
0.59
Measurement: solid line, Estimate: dashed line
p/ρU²
Time (s)
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
86
88
90
92
94
96
98
100
102
p1
p3
p11
p16
Measurement: solid line, Estimate: dashed line
92
94
96
98
100
102
104
106
108
110

(a) $k = 1$

(b) $k = 9$



(c) $k = 18$

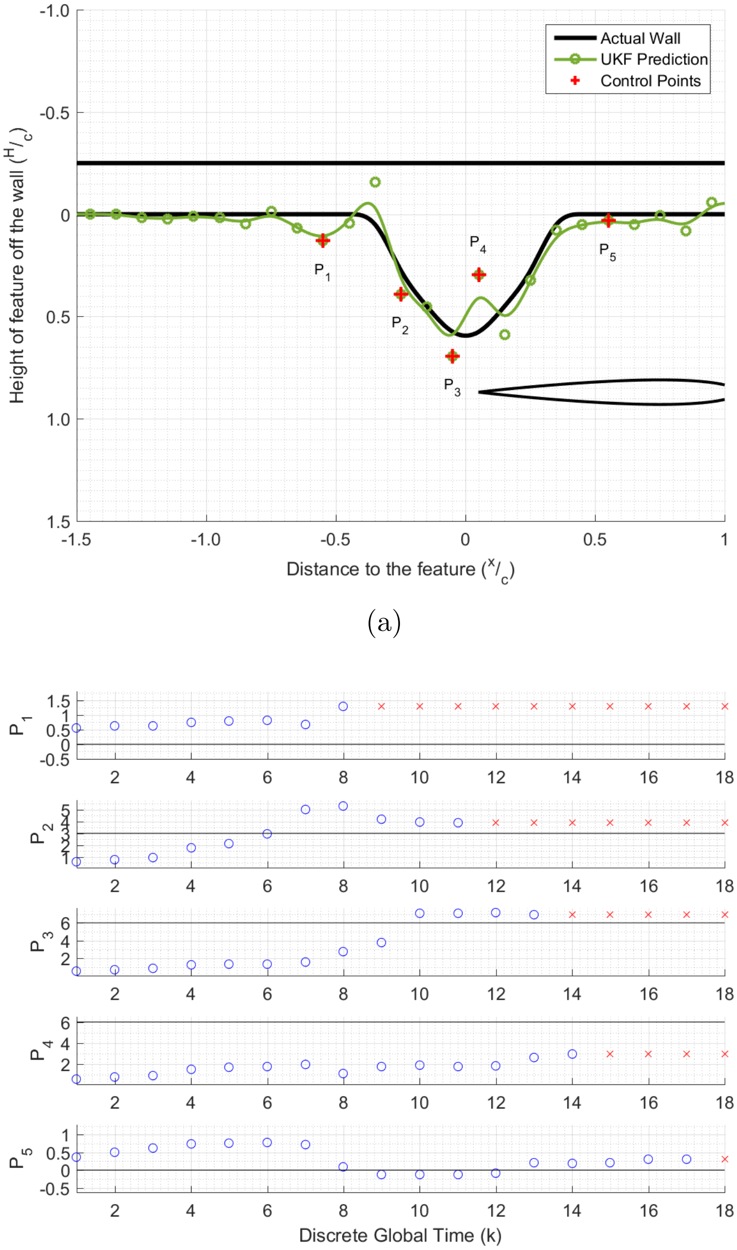
Figure 25: Pressure time histories corresponding to three timesteps of the *UKF*. Actual (top) and estimated (bottom) simulations are shown in the upper left corner with vorticity.

While the general shape of each pressure time history is nearly matched by the estimate, small deviations from the true values lead to minor adjustments in the control points, which will alter the shape estimate. While this method can estimate general boundary conditions from simulated pressure time histories, in a practical sense, it would be necessary to minimize these error. Downstream errors in the wall estimate will induce errors in upstream estimates that will propagate error in future estimates.

A significant portion of the observed error between pressure measurements appears as a time lag in the estimate. One approach to addressing this error would be slowing the forward speed of the foil, such that changes in the wall geometry are measured more gradually. A second method for addressing the time lag would be to phase shift the pressure measurements based on the lag.

5.3 Control Points

It has been demonstrated that the spline curve estimate from the weighted average method provides a good representation of the wall with relatively low error, with reasonable approximation of the true pressure time histories. However, in the weighted estimate of control points locations, visual inspection indicates that some control point locations are better estimated than others. Observing the vertical position of a select number of control points over time allows insight on the performance of the method in specific regions of the shape. Figure 26a shows the weighted average estimate with five labeled control points shown with a red $+$ and labeled $P_1, \ldots, P_5$, and Figure 26b shows the estimates of the positions of those five control points over time. The vertical locations of control points are well estimated for $P_2,\ P_3,$ and P_5. Figure 26b shows how these three control point estimates move towards the true locations and stay close to those positions until they are frozen. However, P_1 and P_4 deviate significantly from their true locations, either diverging from the start or never moving far enough. This could be due to over-fitting the spline with control points, choice of initial conditions, or the control point estimates becoming stuck in a local minimum. In the case of over-fitting, the estimate could benefit from removing excess control points or “thinning” of the spline (Figure 27).



(a)

(b)

Figure 26: Five control points over the global time k for the weighted average estimate. Control points labeled with a red + in (a) correspond to the plots in (b). In (b), blue Os mean control points are still in the window, red Xs mean the window has passed and that is now the fixed location of the control point. Each subplot in (b) has a black line to indicate the true value of that control point.

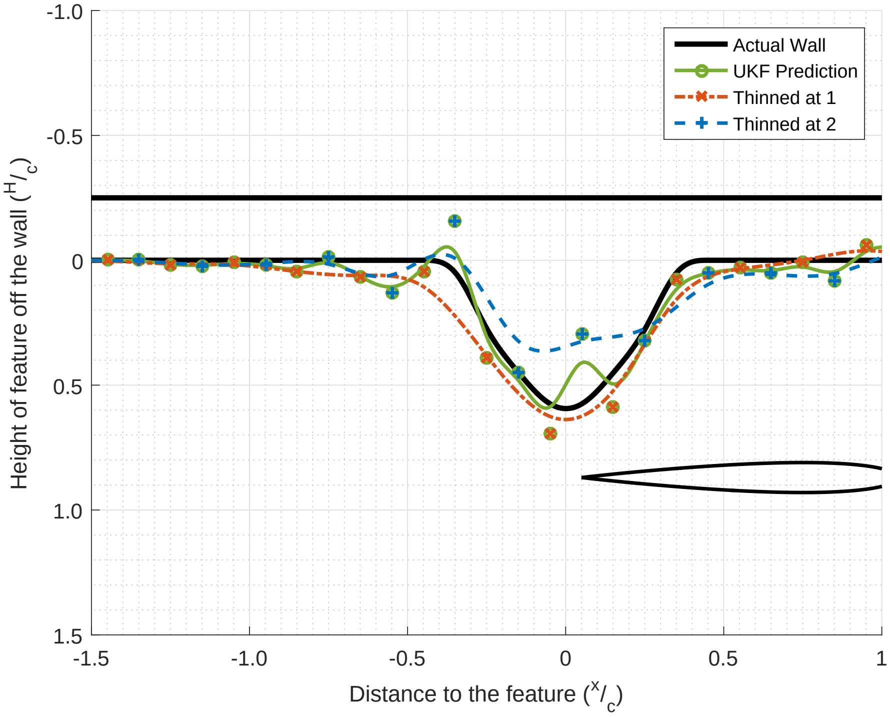


Figure 27: How removing every other control point from the estimate affects the spline starting with the first (orange) and second (blue) control point. Control points are shown with shapes (O, X, +) and colored to correspond with its curve.

Figure 27 shows that the shape estimate improves by removing some of the control points. When starting at the first control point and thinning every other control point (orange dashed line in Figure 27) it can be seen that the spline estimate comes much closer to the true shape without fluctuations. On the contrary, starting at the second control point (blue dashed-dot line in Figure 27) and removing every other point, the poorly estimated points are kept and the estimate is poor. The specific choice in which control points are removed has a strong effect on the curve. Since the true shape is unknown in a given estimate, there is no way too determine which control points may be considered poor estimates and a different method is necessary if one wants to improve the estimate of the shape.

5.4 A Second Pass

Selection of initial conditions is critically important for any Bayesian estimation problem, as the estimate is predicated on the refining an a priori estimate. Maertens [46] demonstrated that performing a passing the object a second time using the previous estimate as the initial conditions can improve the estimate of fine features. Following this, the weighted average estimate was used as the initial conditions for a second pass (Figure 28). The second pass was performed under two circumstances: 1) using the unmodified control points from the weighted average estimate as the initial conditions and 2) using the weighted average estimate as the initial conditions but modifying the three control points known to be poor estimates.

Figure 28 shows the original weighted average estimate as well as the unmodified and modified second pass. When initializing the method with the previously estimated control points, the overall estimate does not change substantially except in the region close to the dip in the peak of the estimate. In this region, the poorly estimated control point (P_4) causes confusion in the solution and drives the neighboring point further away from its true position.

Instead, modifying the three control points that are known to be poor estimates improves the overall estimate of the wall. Here it can be seen by visual inspection that the estimate has smaller fluctuations on the flat wall upstream of the feature and more closely matches the width, height, and location. This shows the importance of initial conditions and highlights the sensitivity of each control point to the location of its neighbors. When control points are estimated far from their true position it also diverges the neighboring control points. This divergence is likely due to the placement of a control point on the wrong side of a concavity, which due to the design of B-splines, causes substantially different curves. Using pressure as a proxy, it can be difficult to determine which of the $m + 1$ local control points in the window of N control points is causing this deviation.

While this shows that it is possible to improve the estimate with choice of initial conditions and a second pass, it is not always physically possible to decide which control points are poor estimates. If one wants to attempt to decide which control points to remove, observing the vertical change in control point location over the window may provide information about the quality of the estimate, however this would require some previous knowledge of the concavity of the true wall.

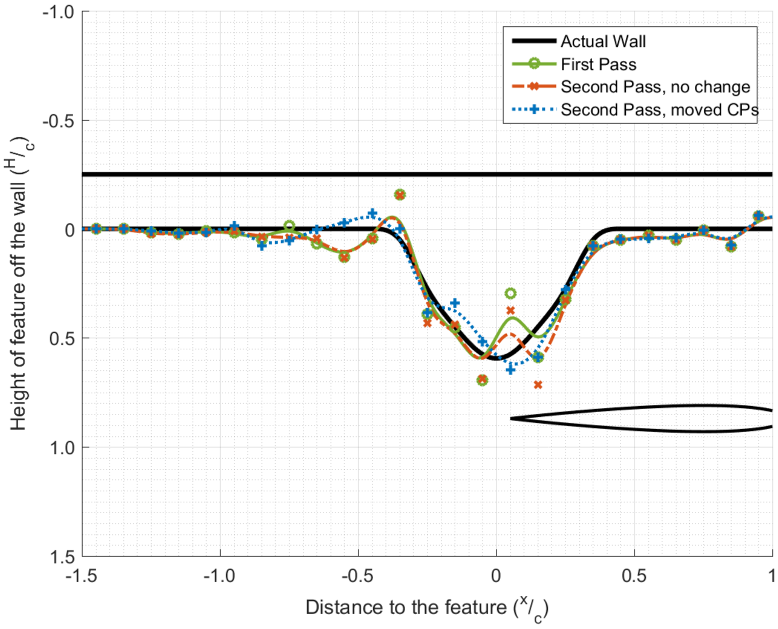


Figure 28: How initializing the method with the weighted average estimate affects the final solution. The first pass (green solid line), the unmodified second pass (orange dashed-dot line), and the modified second pass (blue dotted line) are shown with control points (O, X, +) colored to correspond with each curve.

5.5 Number of Control Points

Another factor in the developed algorithm is the number of control points estimated in a windowed region of size N near the foil body. A small study was performed to vary this number to see the effect on the solution. Figure 29 shows the true wall shape as well as wall estimates for four different window sizes, indicated with different colors and line styles. In each test, a different number of windowed control points N is used, with values ranging from 19 to 7.

Figure 29 shows that increasing the size of the estimation window improves the estimate of the wall. This is likely due to the fact that points on the upstream side of the window are included earlier in time and thus affect the total estimate for a longer period of time. This implies that the speed at which the body moves relative to the window size and the size of the feature is also critically important and could significantly impact the estimate.

A value of $N = 19$ creates a window that is approximately twice the chord length of the foil. This placed control points on the extreme limits of the sensing range discussed by Bouffanais et al. [32]. In contrast, when $N = 7$, control points are very close to the body in the window region such that there is limited opportunity for the algorithm to allow changes to the control point locations before the estimate becomes frozen. As the window size increases, and the estimate was allowed to move control points at the extreme of the sensing range, the estimate better approximates the true solution.

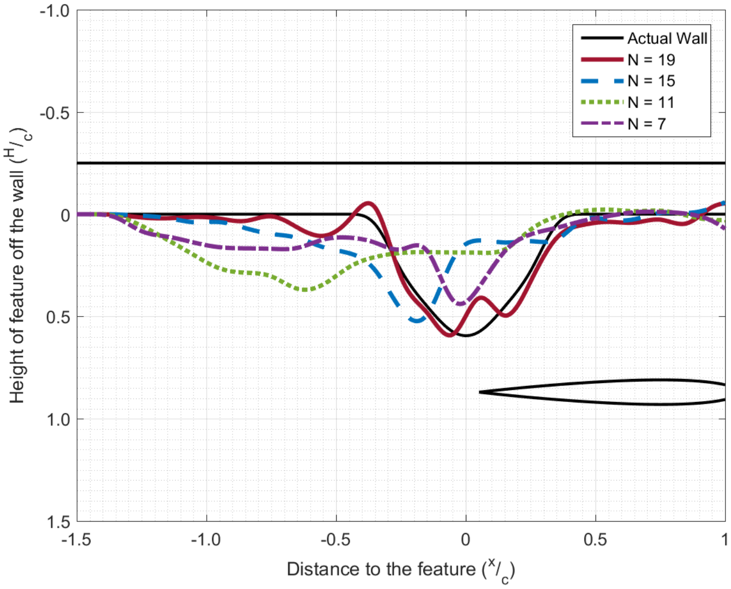


Figure 29: The wall estimate as the size of the windowed region near the foil is modified. Three window sizes are shown, represented by lines of different color and style.

5.6 Effect of location and number of pressure measurements

The number of pressure measurements used for estimating the wall shape can also have a significant effect on the resulting estimate. This is important for practical implementation of the algorithm in a physical system. Selecting points along the foil that provided good information about the feature can improve the accuracy of the estimate while lowering the overall computational costs. Figure 30 below shows how the estimate is affected by using four different combinations of pressure locations along the body of the foil. The four cases are shown in different colors and line styles. A foil body with numbered sensor locations has been placed in the plot to correspond to the locations mentioned in the legend.

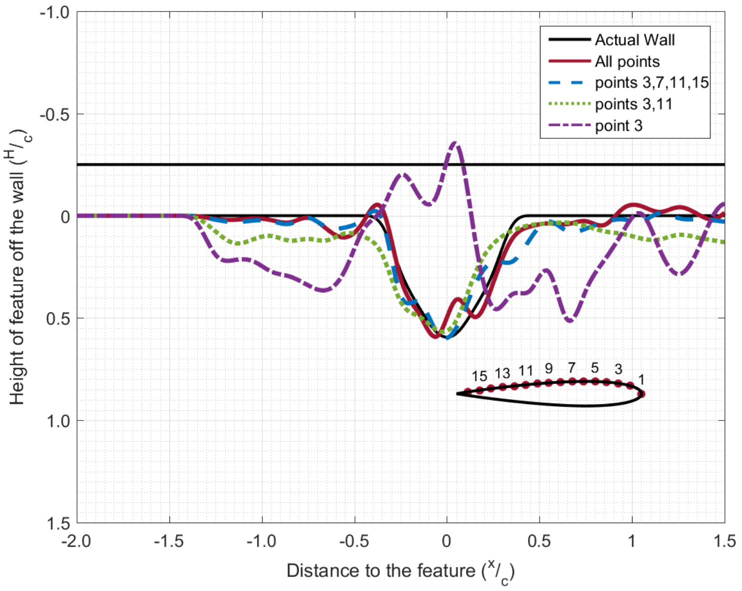


Figure 30: The wall estimate as the number and location of pressure measurements was modified. Pressure locations are read such that the first point is at the leading edge (front) and the 16^{th} point is at the trailing edge (back). Every other pressure sensor along the foil was given a number to help with understanding the legend.

Figure 30 shows that there must be a minimum of two pressure measurements along the length of the foil in order to estimate the shape. Interestingly, it appears that even with a single measurement of pressure, it was possible to determine that a protrusion exists however the estimate of the whole wall appears to be inverted from the true shape. Additionally, as long as there is a measurement of pressure in the wake and near the front of the foil, the feature is distinguished. Estimation of the flat wall regions were similar for four and 16 pressure measurements. While the estimate was more accurate using all measurements of pressure, it is not necessary to use this large number of measurements to estimate the shape. Conversely, it may be necessary to use more measurements of pressure along the length of the body if one wants to recover the exact shape.

5.7 Importance of Measurement Noise

Kalman Filters are predicated on the fact that both the measurement and the state are inherently noisy systems. Due to this, the selection of the noise level is critically important for the success of estimation. In the present study, all measurements are generated through simulated systems, and thus the noise on the measurement and state were assumed to be the resolution level of LilyPad and Matlab respectively. For a physical system, it would be necessary to first characterize the noise levels of the individual sensors, as well as in the entire system, to properly model the noise. It may also be necessary to allow the noise on each sensor as a state of the system, as the noise level could change based on the speed of the body, the characteristics of the fluid, or noise from things outside the system. While this would increase the number of active sigma points, it would guarantee that the noise in the model remains relevant to the physical system.

5.8 How α, β, and κ Affect Estimation

The design of a *UKF* is sensitive to the values chosen for α, β, and κ since they control the approximation of the statistical distributions for the estimated random variables. These values must be "tuned" a priori for a given system to optimize the performance of the estimator. The most important constant is α, as it controls the spread of sigma points around the current estimated state and also affects the magnitude of the weights used in the unscented transform. Small values of α will allow only small variability about the current estimated state which may result in difficulty in capturing the true statistics of the system. Further, when the value of α is small, the weights for the unscented transform are large, which can provide heavy weight to a poor solution. On the contrary, when α is too large, it is possible that the true state of the system will be difficult to estimate due to too much variability allowed in determining sigma points, though the weights will be lower and the risk of applying heavy weight to a poor solution is smaller.

Figure 31 shows how α affects the estimation of the shape, using small numbers as suggested by Wan and Van Der Merwe and then values closer to those used by Julier [45]. Values of α are shown as different line colors and styles. For $\alpha = 0.01$, the estimate is very poor as the control point location distributions are too narrowly constrained, while for $\alpha \geq 1$ the estimation of the shape is close to the true shape and relatively unchanged from one estimate to the other. However, it can be seen here that there is significant deviation from the flat wall in estimate on the front side of, and past the shape for $\alpha = 3$ with similar but smaller deviations seen for $\alpha = 2$.

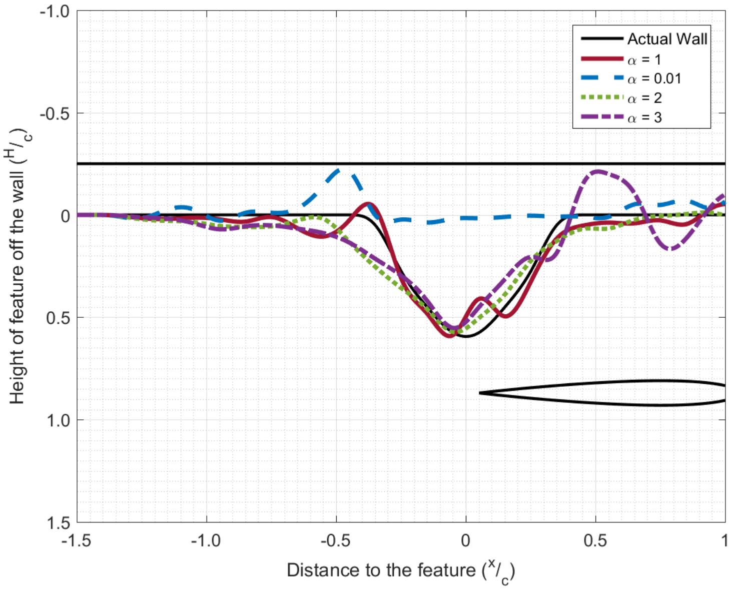


Figure 31: A plot showing the estimate of the wall for varying values of α with the original estimate presented previously ($\alpha = 1$). The different predictions are denoted with both color and line type to set them apart from the true wall.

When compared to the effect of α, changing β and κ has significantly smaller effect on the estimates. Modifying β indirectly affects the solution by modifying the covariance matrix to incorporate higher order information about the distributions of the random variables being estimated. Figure 32 shows how the solution changes slightly with modification of β. One interesting observation is that the spline approximation appears to improve minimally with increased β. While $\beta = 2$ corresponds to a Gaussian distribution for the control point locations, increased values of β result in increasingly non-Gaussian distributions, showing that the assumption of a Gaussian distribution may not be best for these random variables.

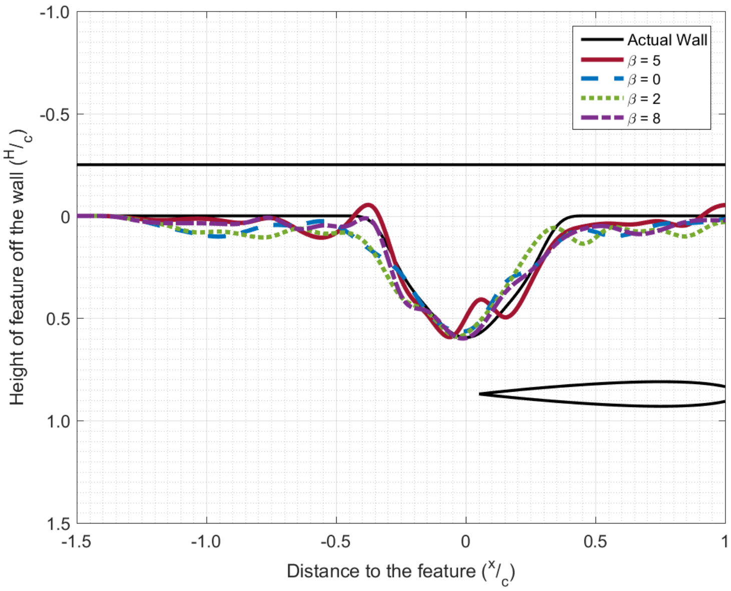


Figure 32: A plot showing the estimate of the wall for varying values of β with the original estimate presented previously ($\beta = 5$). The different predictions are denoted with both color and line type to set them apart from the true wall.

κ also has a small effect on the estimation and it was often suggested in the literature to set κ as 0 or 1 unless the covariance matrix of the state estimate is forced away from being positive definite [42, 45]. In Figure 33 one sees that increasing the value of κ can greatly influence the estimate in a negative way. As the value of κ increases, the estimate on both the front and the back of the feature diverges from the true shape. Increasing κ can increase fluctuations in the flat wall estimate, as seen to a severe degree for $\kappa = 5$.

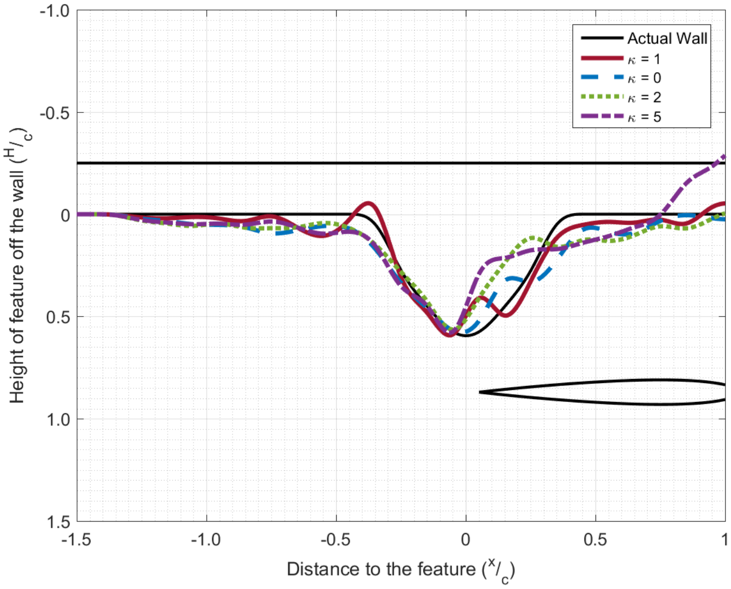
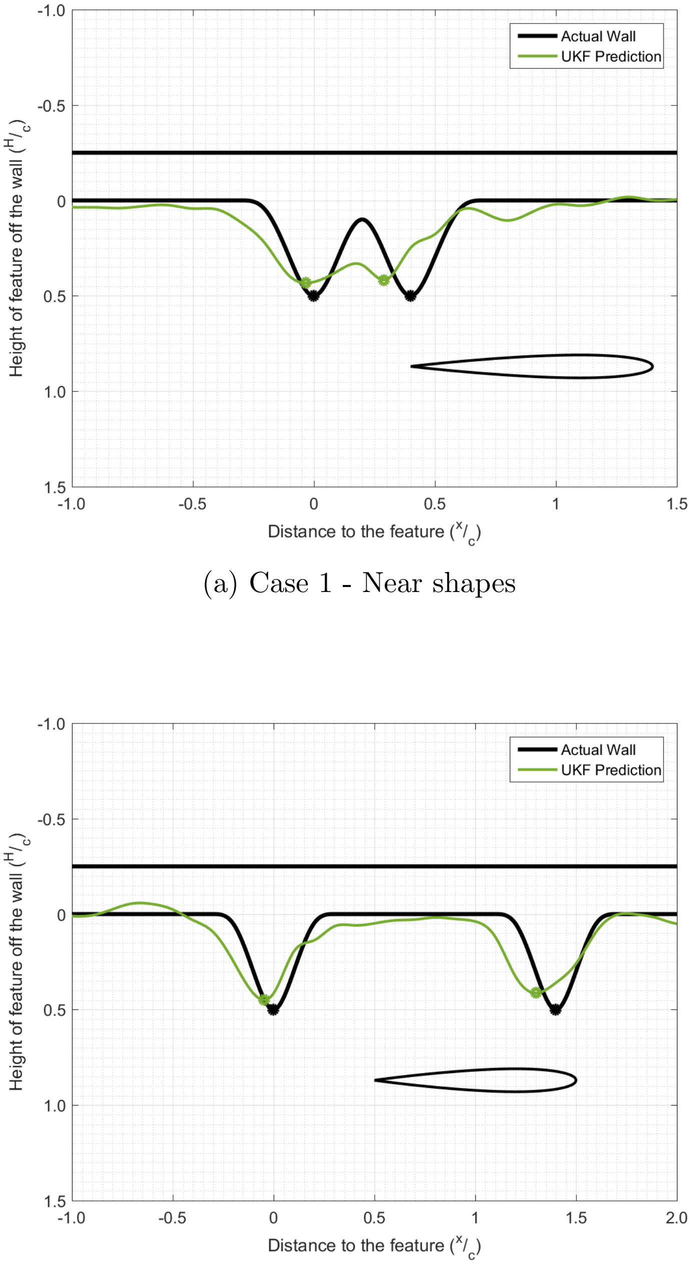


Figure 33: A plot showing the estimate of the wall for varying values of κ with the original estimate presented previously ($\kappa = 1$). The different predictions are denoted with both color and line type to set them apart from the true wall.

5.9 Multiple Shapes

The weighted estimation algorithm is tested with multiple shapes to investigate the estimation method under more general circumstances. Here, two features protrude from the wall in the vicinity of the foil, and the distance between the protrusions is varied. In particular there is interest in: 1) how the estimate performs with a wall that has multiple features and 2) how having features close to one another can create ambiguity in the estimate. For comparison, we define a case where the features are close to one another (Figure 34a) and spread further apart (Figure 34b). Figure 34 shows the final wall estimates for both cases using the weighted average *UKF*, with circles indicating the estimated peaks of each feature. Table 9 presents the performance analysis for both the near and far cases.



(a) Case 1 - Near shapes

(b) Case 2 - Far shapes

Figure 34: An estimate of wall shape with two features placed in the simulation. Circles indicate the location of the peaks of the feature.

Figure 34 shows that it is possible to estimate multiple features with a single pass of the foil body, though there are some limitations to the method. This is evidenced in Figure 34a where the height and width of the overall feature is estimated reasonably well, however the true shape is not distinguishable, with some blending of the two features. This is likely due to the formation of a recirculation region between the peaks of the two features, creating a situation where pressure may not be able to distinguish between solid walls and a streamline between peaks. In contrast, moving the shapes apart allows far less ambiguity in the wake such that information about the second peak is captured, as seen in Figure 34b. In this example, the a separation distance was approximately one chord length of the foil, such that the second shape was not passed by the foil until the first shape had already been passed with an estimate frozen in place. In this case, the size and location of both features was reasonable, though the second shape is estimated to be wider and shorter than the true feature.

Contrary to the visual inspection in Figure 34, where it appears the estimate of case 2 is better, Table 9 shows that the two cases are similar in terms of error. In case 1, where the two shapes are near to one another, it was difficult to distinguish between the two features in the estimate and the errors were relatively high for both features. Interestingly, when the features are separated as in case 2, the first feature is estimated with relatively low error and the second feature becomes more difficult to estimate. This in somewhat expected, as any errors in the estimate of first feature will propagate in the simulations used to estimate the second feature, inducing additional errors. Overall, while it is possible to distinguish between multiple shapes it becomes more difficult to determine the true shape due propagation of error.

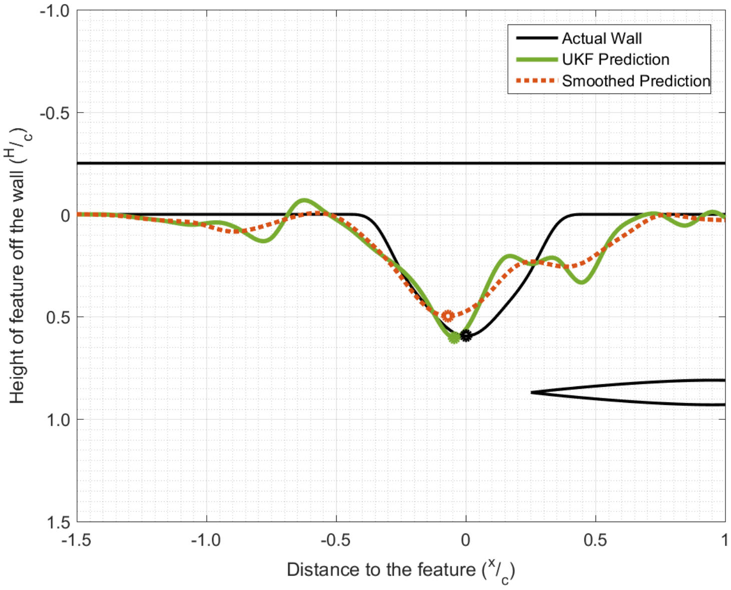
Table 9: Error metrics for both multi-shape cases. Height, width, and distance are in units of chord lengths, and the center is defined by the physical location of the center referenced from the start position of the simulations. Each feature is presented separately, while RMSE_T was common between both features. The characteristics of the true wall are also presented for comparison.

Method	H	W	dis	Cen	$\tilde{H}$	$\tilde{W}$	$\tilde{dis}$	$\tilde{Cen}$	RMSE_F	RMSE_T
Case 1 (1^{st})	0.50	0.23	0.25	101.0	-	-	-	-	-	-
Estimate	0.43	0.38	0.32	100.7	0.13	0.64	0.27	0.003	0.061	0.042
Case 1 (2^{nd})	0.50	0.23	0.25	105.0	-	-	-	-	-	-
Estimate	0.42	0.27	0.33	103.9	0.17	0.17	0.33	0.010	0.178	0.042
Case 2 (1^{st})	0.50	0.23	0.25	100.0	-	-	-	-	-	-
Estimate	0.45	0.30	0.30	99.5	0.10	0.27	0.20	0.005	0.097	0.043
Case 2 (2^{nd})	0.50	0.23	0.25	114.0	-	-	-	-	-	-
Estimate	0.41	0.39	0.34	113.0	0.17	0.67	0.35	0.009	0.100	0.043

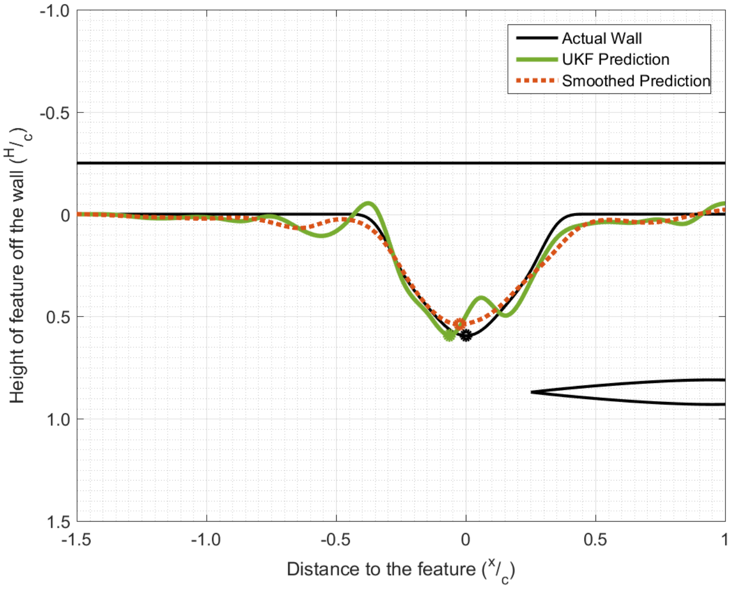
5.10 Filtering

Since only a few control points are observed to be poor estimates, a smoothing method is used to blend poor control points with good ones and improve the overall estimate. A Savitzky-Golay moving average filter blends control point estimates in a local region, removing small scale fluctuations [47]. This type of filter was applied to the final estimates of wall shapes in the previously tested examples to evaluate how smoothing can improve the estimates.

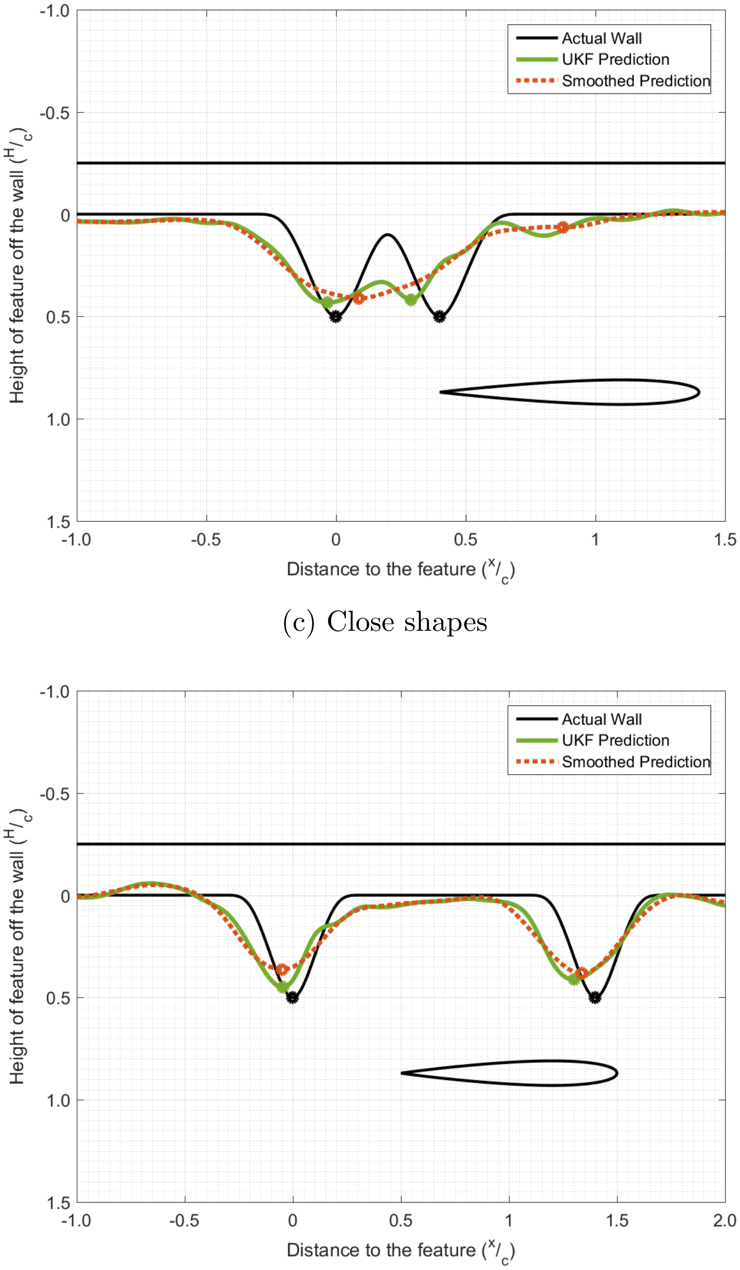
Figure 35 shows a comparison between the true wall shape (black), the final wall estimate from the *UKF* (orange), and the estimate after applying the smoothing filter (green) for the four presented cases with circles indicating each estimated peak The estimate of the peak(s) of the features were taken as the point(s) closest to the foil body within the feature. Error metrics showing the performance of the smoothing filter are seen in Table 10, where each divided row corresponds to a previously tested example. Each row of Table 10 presents the true feature characteristics, the original estimate from the *UKF*, and the estimate after smoothing (in bold text).



(a) The standard measurement design



(b) The weighed average measurement design



(c) Close shapes

(d) Far shapes

Figure 35: Smoothing the estimates from the b-spline *UKF*. The true wall is shown (black) as well as the estimate from the *UKF* (orange) and the smoothed solution (blue). Circles with colors corresponding to each wall shape are shown to indicate the location of the observed peak(s).

Table 10: Error metrics for the smoothed curves. Height, width, and distance are in units of chord lengths, and the center is defined by the physical location of the center referenced from the start position of the simulations. True characteristics of the spline curves and the original estimates are given for reference, with the smoothed results given in bold.

Method	H	W	dis	Cen	$\tilde{H}$	$\tilde{W}$	$\tilde{dis}$	$\tilde{Cen}$	RMSE$_F$	RMSE$_T$
Standard	0.59	0.48	0.16	102.5	-	-	-	-	-	-
Original	0.60	0.69	0.15	103.9	0.013	0.44	0.05	0.014	0.126	0.063
Smoothed	**0.50**	**0.81**	**0.25**	**96.1**	**0.17**	**0.69**	**0.63**	**0.062**	**0.105**	**0.523**
Weighted	0.59	0.48	0.16	102.5	-	-	-	-	-	-
Original	0.59	0.53	0.16	103.0	0.003	0.12	0.01	0.005	0.076	0.026
Smoothed	**0.54**	**0.52**	**0.21**	**97.5**	**0.09**	**0.09**	**0.34**	**0.049**	**0.034**	**0.021**
Case 1 (1^{st})	0.50	0.23	0.25	101.0	-	-	-	-	-	-
Original	0.43	0.38	0.32	100.7	0.13	0.64	0.27	0.003	0.061	0.042
Smoothed	**0.41**	**0.70**	**0.34**	**101.9**	**0.17**	**2.0**	**0.35**	**0.009**	**0.076**	**0.044**
Case 1 (2^{nd})	0.50	0.23	0.25	105.0	-	-	-	-	-	-
Original	0.42	0.27	0.33	103.9	0.17	0.17	0.33	0.010	0.178	0.042
Smoothed	**0.00**	**0.00**	**0.75**	**-**	**1.0**	**1.0**	**1.0**	**-**	**-**	**0.044**
Case 2 (1^{st})	0.50	0.23	0.25	100.0	-	-	-	-	-	-
Original	0.45	0.30	0.30	99.5	0.10	0.27	0.20	0.005	0.097	0.043
Smoothed	**0.38**	**0.43**	**0.39**	**99.5**	**0.24**	**0.86**	**0.47**	**0.005**	**0.099**	**0.044**
Case 2 (2^{nd})	0.50	0.23	0.25	114.0	-	-	-	-	-	-
Original	0.41	0.39	0.34	113.0	0.17	0.67	0.35	0.009	0.100	0.043
Smoothed	**0.36**	**0.42**	**0.37**	**113.4**	**0.28**	**0.82**	**0.55**	**0.006**	**0.100**	**0.044**

Figure 35a shows the smoothing filter applied to the initial standard update formulation feature estimate. It can be seen that if the estimate of the wall has significant deviations from the true solution, it can be difficult or impossible to smooth away the undesirable fluctuations in the solution. Further, deviations in height on the back side of the foil appear to push the estimate of the peak away from the true value through smoothing. In contrast, when the overall fluctuations from the true solution are small, smoothing appears to improve the estimate of the wall as seen in Figure 35b, where the filter is applied to the weighed average solution. Here, the peak is not significantly affected by smoothing and appears to move closer to the true location in the horizontal direction, although the height of the feature is reduced due to the dip in the solution near the peak. Fluctuations on the flat wall portions of the estimate are reduced and the deviation near the peak is removed through smoothing.

Figure 35c shows the smoothing filter applied to the estimate of two features close to one another. In the case of two close features, smoothing removes information about the second feature and instead blends the two features together. Smoothing the solution now provides an estimate of only a single feature with width close to that of the two shapes combined and height close to that of the protrusions. Further, there is a second estimated “peak” on the back side of the feature, a result of poorly smoothing the second feature with estimated fluctuations on the flat wall.

Figure 35d shows the smoothing filter applied to the estimate of two features far from one another. In this case, if the estimate is already relatively smooth, it is difficult to remove the undesirable parts of the solution via through the smoothing filter. Smoothing the estimate of the two separated shapes reduces the magnitude of the height of the estimated peaks and does little to remove deviations from the flat wall, though the first peak is a much smoother curve. While the estimate of the two shapes was previously deemed to be adequate, smoothing does not greatly improve the accuracy of the wall estimate, and can cause an underestimate of the height of the wall.

Smoothing generally reduces the magnitude of the estimate of the peak(s) of the features being estimated and can cause an over estimate of the width; however, when the estimate of the wall is good as in Figure 35b, smoothing can reduce the magnitude of the fluctuations in the flat wall region and can slightly improve the estimate. This is important if one wanted to use this method to update boundary conditions in a digital twin, where proper local curvature of the boundaries could have a significant effect on the numerical solution.

These trends are emphasized in Table 10 where smoothing only improves results for the singles feature weighted method solution. For the standard update formulation, smoothing caused increases in percent errors between 2% and 58%, and increased the total curve RMSE by ∼700%. The weighted average solution shows reduced errors in specific characteristic estimates such as width, the total RMSE, and the feature RMSE, while other error values are increased by a significant margin. The minimum distance error increased from 1% to 34%, the error in the height was changed from 0.3% to 9%, and the error in the estimate of the center was increased from 0.5% to 5%. Further, Table 10 shows that for both cases involving multiple feature, the error either increased or remained approximately the same. Case 1 and case 2 in Table 10 demonstrate that if the case does not present with low initial error, smoothing can accentuate error rather than improve the solution.

CHAPTER 6

Conclusions

This thesis investigates the development of an estimation method for determining general wall features that can be used for object identification or updating the general solid boundary conditions in a digital twin. Based on the receptors of the mechanosensory lateral line system, two methods were investigated for the development of the estimation algorithm. A method of classification via neural networking and machine learning was developed using pressure time histories from LilyPad to infer the class of the wall in a simulation. Also investigated was a method to estimate general boundary conditions with B-splines using an Unscented Kalman filter and pressure measurements as a proxy. The investigation of the two methods were performed using the 2D viscous flow solver LilyPad as a model, to determine the relationship between pressure and estimates.

The first method demonstrated potential for the purposes of this investigation, showing low error in the prediction of wall classes. Further, as the level of ambiguity in the available classes was reduced, the errors in prediction shifted from shape misclassification to misclassification of size. Though this showed that removing ambiguity from the solution improved classification, it would require a nearly infinite training set in order to make general classifications. While it would be prohibitive to generate this training set for all possible features, the machine learning method could be used as a secondary system to classify object features which have already been estimated. In this way, the classifier would provide secondary details about features based on a more general characterization and the pressure time history.

For the second method investigated, it was found that the algorithm can be used for the estimation of general boundary conditions which govern the shape and location of features in a region local to the foil based on measurements of pressure at the surface of the foil. Representing the wall with B-spline control points, estimated using an Unscented Kalman filter, it was possible to estimate the general shape. A weighted average method was presented for the state update stage of the filter, which showed improvement over the standard formulation of the *UKF*. The weighted average method showed improvements for both fine features, and estimation of flat wall regions which lie on the edge of the sensing range of the body. It was also shown to reduce all error metrics used to define the performance of the method.

While the method is able to reliably recognize the location of features and does a good job of determining shape and size, it is sensitive to the selection of the *UKF* constants α, β, and κ. It is critically important for the quality of the estimate that an adequate number of pressure measurements, an appropriate window size, and a sufficient number of control points was selected such that the true characteristics of the feature are captured by the estimated signal.

Also demonstrated was the ability of the method to identify multiple features on a wall, though bringing the shapes too close caused the system to estimate a single feature. Here, it was difficult to determine features that were too close due to the region between the peaks of the features over which the wake of the body did not have time to respond to the second feature. Conversely, when there is sufficient spacing between the features such that the first feature is frozen in place during the estimation of the second feature, the estimate of the second feature improves.

CHAPTER 7

Suggested Future Work

- Increase the resolution of the simulation in both time and space such that the solution more accurately captures the physics of the system. A simulation resolution on the order of 2^9 or 2^{10} with a foil body resolution of at least 20 grid points should provide more information at every time step, as well as finer information that was not available in the previously coarse mesh. Using this method, it would also be reasonable to measure pressure at more locations along the length of the foil.

- Investigate different estimation windows and how the size and shape of the feature of interest can influence the required window size.

- Test the influence of the constants α, β, and κ for a variety of more generalized shapes than those used in this investigation, it an attempt to determine globally default values for each constant.

- Investigate a more dynamic definition of α such that the spread of the sigma points was determined in-situ at each iteration based on the quality of the previous estimate.

- The error space of the estimate window should be observed to determine the influence each control point (relative to its position in the window) has on each measurement of pressure on the body. This information could be used to modify the weighting step such that each control point in a window estimate corresponding to a specific pressure measurement is weighted on its relative affect on that pressure measurement location, and then the whole window is weighted as previously mentioned.

- Perform an analysis of the propagation of downstream errors in the estimate, in order to quantify the influence of mis-calculated control points on upstream estimates. Errors in the downstream control points influence the location of upstream control points due to induced errors in the shape of the pressure measurement.

- LilyPad can be initialized with a field of vorticity, such that rather than starting all simulations at some global zero time, simulations used for the unscented transform could be initialized with the previous flow field at a time much closer to the current time k. While this would lower the computational time, it is possible that this method could induce errors through initialization of a flow field that does not exactly match the flow field at the end of the previous run.

- Due to its design, the presented method could be extended to the estimation of a series of 3-D control points which define a Non-Uniform Rational Basis Spline (NURBS) surface. This would require implementation with a 3-D viscous model and a new formulation of the state equation (Equation 4) such that it corresponds with the new viscous model.

- B-spline curves are determined by the position of control points and a basis matrix $B_{i,m}(x)$. It would be of interest to see if using a different basis matrix formulation changes the solution.

LIST OF REFERENCES

[1] H. Bleckmann and R. Zelick, “Lateral line system of fish,” *Integrative zoology*, vol. 4, no. 1, pp. 13–25, 2009.

[2] J. F. Webb and J. B. Ramsay, “New interpretation of the 3-d configuration of lateral line scales and the lateral line canal contained within them,” *Copeia*, vol. 105, no. 2, pp. 339–347, 2017.

[3] J. F. Webb, “Lateral line structure,” *Encyclopedia of fish physiology: From genome to environment*, vol. 1, pp. 336–346, 2011.

[4] S. Coombs and S. Van Netten, “The hydrodynamics and structural mechanics of the lateral line system,” *Fish physiology*, vol. 23, pp. 103–139, 2005.

[5] B. M. Howe, L. Van Uffelen, E.-M. Nosal, and G. S. Carter, “Acoustic seagliders in philsea10: Preliminary results,” in *OCEANS 2011*. IEEE, 2011, pp. 1–8.

[6] D. Roemmich, G. C. Johnson, S. Riser, R. Davis, J. Gilson, W. B. Owens, S. L. Garzoli, C. Schmid, and M. Ignaszewski, “The argo program: Observing the global ocean with profiling floats,” *Oceanography*, vol. 22, no. 2, pp. 34–43, 2009.

[7] C. Roman, G. Inglis, and J. Rutter, “Application of structured light imaging for high resolution mapping of underwater archaeological sites,” in *OCEANS 2010 IEEE-Sydney*. IEEE, 2010, pp. 1–9.

[8] S. Boschert and R. Rosen, “Digital twinthe simulation aspect,” in *Mechatronic Futures*. Springer, 2016, pp. 59–74.

[9] E. J. Tuegel, A. R. Ingraffea, T. G. Eason, and S. M. Spottswood, “Reengineering aircraft structural life prediction using a digital twin,” *International Journal of Aerospace Engineering*, vol. 2011, 2011.

[10] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui, “Digital twin-driven product design, manufacturing and service with big data,” *The International Journal of Advanced Manufacturing Technology*, vol. 94, no. 9, pp. 3563–3576, Feb 2018. [Online]. Available: https://doi.org/10.1007/s00170-017-0233-1

[11] R. Söderberg, K. Wärmefjord, J. S. Carlson, and L. Lindkvist, “Toward a digital twin for real-time geometry assurance in individualized production,” *CIRP Annals*, vol. 66, no. 1, pp. 137–140, 2017.

[12] E. Glaessgen and D. Stargel, "The digital twin paradigm for future nasa and us air force vehicles," in *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, 2012, p. 1818.

[13] A. Ramtilak, A. Joseph, G. Sivakumar, and S. Bhat, "Digital twin spark ignition for improved fuel economy and emissions on four stroke engines," SAE Technical Paper, Tech. Rep., 2005.

[14] D. Renzi, D. Maniar, S. McNeill, C. Del Vecchio, *et al.*, "Developing a digital twin for floating production systems integrity management," in *OTC Brasil*. Offshore Technology Conference, 2017.

[15] S. Dijkgraaf, "The functioning and significance of the lateral-line organs," *Biological Reviews*, vol. 38, no. 1, pp. 51–105, 1963. [Online]. Available: http://dx.doi.org/10.1111/j.1469-185X.1963.tb00654.x

[16] B. L. Partridge and T. J. Pitcher, "The sensory basis of fish schools: Relative roles of lateral line and vision," *Journal of comparative physiology*, vol. 135, no. 4, pp. 315–325, Dec 1980. [Online]. Available: https://doi.org/10.1007/BF00657647

[17] J. Engelmann, W. Hanke, J. Mogdans, and H. Bleckmann, "Neurobiology: hydrodynamic stimuli and the fish lateral line," *Nature*, vol. 408, no. 6808, p. 51, 2000.

[18] S. Coombs, C. B. Braun, and B. Donovan, "The orienting response of lake michigan mottled sculpin is mediated by canal neuromasts," *Journal of Experimental Biology*, vol. 204, no. 2, pp. 337–348, 2001.

[19] J. F. Webb, N. C. Bird, L. Carter, and J. Dickson, "Comparative development and evolution of two lateral line phenotypes in lake malawi cichlids," *Journal of morphology*, vol. 275, no. 6, pp. 678–692, 2014.

[20] J. Chen, J. Engel, N. Chen, S. Pandya, S. Coombs, and C. Liu, "Artificial lateral line and hydrodynamic object tracking," in *Micro Electro Mechanical Systems, 2006. MEMS 2006 Istanbul. 19th IEEE International Conference on*. IEEE, 2006, pp. 694–697.

[21] J. C. Montgomery, S. Coombs, and C. F. Baker, "The mechanosensory lateral line system of the hypogean form of astyanax fasciatus," in *The biology of hypogean fishes*. Springer, 2001, pp. 87–96.

[22] C. Von Campenhausen, I. Riess, and R. Weissert, "Detection of stationary objects by the blind cave fishanoptichthys jordani (characidae)," *Journal of comparative physiology*, vol. 143, no. 3, pp. 369–374, 1981.

[23] M. Asadnia, A. Kottapalli, D. Karavitaki, M. Warkiani, J. Miao, D. Corey, and M. Triantafyllou, "From biological cilia to artificial flow sensors: Biomimetic soft polymer nanosensors with high sensing performance," *Scientific Reports*, p. DOI: 10.1038/srep32955, 2016.

[24] A. Kottapalli, M. Asadnia, R. Haghighi, A. Cloitre, P. V. y Alvarado, J. Miao, and M. Triantafyllou, "Mems sensors for assessing flow-related control of an underwater biomimetic robotic stingray," *Bioinspiration and Biomimetics*, vol. 10, p. 3, 2015.

[25] A. Kottapalli, M. Asadnia, J. Miao, and M. Triantafyllou, "Artificial fish skin of self-powered micro-electromechanical systems hair cells for sensing hydrodynamic flow phenomena," *J. R. Soc. Interface*, vol. 12, pp. 1–14, 2015.

[26] V. I. Fernandez, S. M. Hou, F. S. Hover, J. H. Lang, and M. S. Triantafyllou, "Development and application of distributed mems pressure sensor array for auv object avoidance," Massachusetts Institute of Technology. Sea Grant College Program, Tech. Rep., 2009.

[27] A. Mivehchi, J. Dahl, and S. Licht, "Heaving and pitching oscillating foil propulsion in ground effect," *Journal of Fluids and Structures*, vol. 63, pp. 174 – 187, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0889974615300657

[28] Y. Yang, J. Chen, J. Engel, S. Pandya, N. Chen, C. Tucker, S. Coombs, D. L. Jones, and C. Liu, "Distant touch hydrodynamic imaging with an artificial lateral line," *Proceedings of the National Academy of Sciences*, vol. 103, no. 50, pp. 18 891–18 895, 2006. [Online]. Available: http://www.pnas.org/content/103/50/18891

[29] R. Venturelli, O. Akanyeti, F. Visentin, J. Jeov, L. D. Chambers, G. Toming, J. Brown, M. Kruusmaa, W. M. Megill, and P. Fiorini, "Hydrodynamic pressure sensing with an artificial lateral line in steady and unsteady flows," *Bioinspiration & Biomimetics*, vol. 7, no. 3, p. 036004, 2012. [Online]. Available: http://stacks.iop.org/1748-3190/7/i=3/a=036004

[30] V. I. Fernandez, A. Maertens, F. M. Yaul, J. Dahl, J. H. Lang, and M. S. Triantafyllou, "Lateral-line-inspired sensor arrays for navigation and object identification," *Marine Technology Society Journal*, vol. 45, no. 4, pp. 130–146, 2011.

[31] A. P. Maertens, J. M. Dahl, and M. S. Triantafyllou, "Distributed pressure sensing to locate and identify obstacles," in 17^{th} *International Unmanned Untethered Submersible Technology (UUST 2011) Conference*, Portsmouth, NH, USA., aug 2011.

[32] R. Bouffanais, G. D. Weymouth, and D. K. Yue, "Hydrodynamic object recognition using pressure sensing," in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences.* The Royal Society, 2010, p. rspa20100095.

[33] S. P. Windsor and M. J. McHenry, "The influence of viscous hydrodynamics on the fish lateral-line system," *Integrative and comparative biology*, vol. 49, no. 6, pp. 691–701, 2009.

[34] A. P. Maertens and M. S. Triantafyllou, "The boundary layer instability of a gliding fish helps rather than prevents object identification," *Journal of Fluid Mechanics*, vol. 757, pp. 179–207, 2014.

[35] L. H. Boulogne, B. J. Wolf, M. A. Wiering, and S. M. van Netten, "Performance of neural networks for localizing moving objects with an artificial lateral line," *Bioinspiration & biomimetics*, vol. 12, no. 5, p. 056009, 2017.

[36] G. D. Weymouth, "Lily pad: Towards real-time interactive computational fluid dynamics," *Numerical Towing Tank Symposium*, vol. 18, oct 2015.

[37] G. D. Weymouth and D. K.-P. Yue, "Boundary data immersion method for cartesian-grid simulations of fluid-body interaction problems," *Journal of Computational Physics*, vol. 230, no. 16, pp. 6233–6247, jul 2011.

[38] Mathworks. "What is machine learning? 3 things you need to know." 2018. [Online]. Available: https://www.mathworks.com/discovery/machine-learning.html

[39] T. Hastie, R. Tibshirani, and J. H. Friedman, *The elements of statistical learning data mining, inference, and prediction*, 2nd ed. Springer, 2011, vol. 1.

[40] E. B. Tchernev, R. G. Mulvaney, and D. S. Phatak, "Investigating the fault tolerance of neural networks," *Neural Computation*, vol. 17, no. 7, pp. 1646–1664, 2005.

[41] C. De Boor, *A practical guide to splines.* Springer-Verlag New York, 1978, vol. 27.

[42] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000.* Ieee, 2000, pp. 153–158.

[43] M. Harashima, L. A. Ferrari, and P. Sankar, "Spline approximation using kalman filter state estimation," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, no. 5, pp. 421–424, 1997.

[44] J. Jauch, F. Bleimund, S. Rhode, and F. Gauterin, "Recursive b-spline approximation using the kalman filter," *Engineering Science and Technology, an International Journal*, vol. 20, no. 1, pp. 28–34, 2017.

[45] S. J. Julier, "The scaled unscented transformation," in *American Control Conference, 2002. Proceedings of the 2002*, vol. 6. IEEE, 2002, pp. 4555–4559.

[46] A. Maertens, "Touch at a distance : underwater object identification using pressure sensors," Thesis (S.M.), Massachusetts Institute of Technology, 2011.

[47] W. H. Press and S. A. Teukolsky, "Savitzky-golay smoothing filters," *Computers in Physics*, vol. 4, no. 6, pp. 669–672, 1990.

APPENDIX A

Explanation of UKF parameters

Table A.1: Relevant parameters to the b-spline *UKF* algorithm with brief explanations of each.

UKF constants	α	Controls the spread of the sigma points
	β	Allows inclusion of higher order statistical moments.
	κ	Maintains positive definiteness of the state covariance.
Spline order	m	Degree of the spline used to calculate the boundary conditions given a set of control points.
Window Size	N	Determines the size of the estimation window near the foil body.
Clamped control points	n	The number of control points to add/remove at every time step of the filter.
Current State	$\bar{\mathbf{P}}$	The current vertical estimate of control points in the window (initially all 0).
State Covariance	C^P	The covariance of the current state (initially an identity matrix).
Sigma Points	$\mathbf{X}$	The L statistically relevant sets of points about the current mean state.
LilyPad	F	The non-linear system equation representing running a LilyPad simulation with a set of control points.
Measurement rate	M	Sets the sample rate of the system.
Measurement array	$\mathbf{p}^{meas}$	The observation time history for the current time step at every location on the foil.
Measurement estimate	$\mathbf{p}^F$	The pressure time histories from LilyPad simulation, run with each set of sigma points.
Unscented estimate	$\mu_{P,F}$	The estimate of the system based solely off the unscented transform of the current state.

APPENDIX B

Description of file structures

The following is a description of the files relevant to this thesis. Bold names indicate folders, italicized names indicate files.

- **B_Spline**:
  - **BezSplineWall_10**: This folder contains the source files for the LilyPad simulations used to generate the sigma simulations
    - **data**: The data folder for LilyPad.exe
      - *done.txt* - This is a flag file generated at the end of a LilyPad run. It tells the ukf that the sigma simulation has finished.
      - *input.txt* - This file contains inputs and can be modified as the simulations are modified. currently: the first line is the foil wall distance. The second is the spline order. The third line are the comma separated x-positions of the control points, The fourth line the y-positions.
      - *output.txt* - the pressure signatures from LilyPad. Column 1 is time, column 2 is not used but corresponds to a previous set of investigations, the rest of the columns are time histories of pressure from the stagnation point to the trailing edge of the foil.
      - *output_dp.txt* - The third column to the end are time histories of the change in pressure between a location on the foil and the stagnation pressure.

    - **LilyPad**: This is the source files, modify these values and then export a new .exe (or your preferred operating system) for use with the ukf.
    - *LilyPad.exe* - An executable from processing to run the LilyPad simulation.
  - **Full_wall**: This contains the observations (pressure), corresponding input files, and the ukf results.
    - **input_files/test_files**: has all the input files/pressure outputs
      - *small_shape_h0p75_furthest_splines.txt* - The standard case used. Presented for column and weighted average method.
      - *small_shape_h0p75_twoPeaks_splines.txt* - The multi-feature case for the close shapes.
      - *TwoSpread_shapes_h0p75.txt* - The multi-feature case for the far shapes.
    - **rolling_ukf**: all the output files from the ukf
      - **dd-mm-yyyy_HH-MM-SS_pressure_plots**: Pressure comparison plots made during method.
      - *dd-mm-yyyy_HH-MM-SS_##CP_initialize.txt(.m)* - The initialization for a run, ## corresponds to the number of CPs used.
      - *dd-mm-yyyy_HH-MM-SS_##CP_data.m* - The data from the ukf. Contains KalFil. Fields: **mu, cpx, mse** structures of the (x,y) points in the window and the rmse for every iteration for each time step. **meas** is the estimated measurement at for each time step. **MSE** is the RMSE at the end of each time step. **xCp, yCp** contains the (x,y) control points for the entire wall estimated thus far at the end of each time step.

  - **SigmaSims**: the Sigma simulations referenced by the ukf. Contains files **BezWall##**, which are files corresponding to the $\#\#^{th}$ sigma simulation. Each file contains a **data** folder, a Java **lib** folder, and the sigma simulation *LilyPad.exe* executable.
  - **Single(three)_ukf**: Results from the ukf before the window.
  - *process_ukf_bsplines.m* - Processed the bspline estimates before the window.
  - *b_spline.m* - A function for generating a basis spline.
  - *rolling_ukf_figures(_type).m* - There are many iterations of this script. They are for generating figures of different types from the outputs of the ukf depending on what type of figure is wanted.
  - *send_shape_ukf.m* - The original ukf method.
  - *bspline_ukf_array.m* - The windowed ukf with measurement as an array.
  - *bspline_ukf_struct.m* - The windowed ukf with the weighted average measurement formulation.
  - *SigmaSimsFileCopy.m* - Used to copy sigma simulations from the source folder to the sigmaSims folder. This is not worth doing by hand.
- **UKF_foil**: The investigation of mean heave distance with LilyPad.
- **UKF_Basic**: Source material about the UKF.

BIBLIOGRAPHY

Asadnia, M., Kottapalli, A., Karavitaki, D., Warkiani, M., Miao, J., Corey, D., and Triantafyllou, M., "From biological cilia to artificial flow sensors: Biomimetic soft polymer nanosensors with high sensing performance," *Scientific Reports*, p. DOI: 10.1038/srep32955, 2016.

Bleckmann, H. and Zelick, R., "Lateral line system of fish," *Integrative zoology*, vol. 4, no. 1, pp. 13–25, 2009.

Boschert, S. and Rosen, R., "Digital twinthe simulation aspect," in *Mechatronic Futures.* Springer, 2016, pp. 59–74.

Bouffanais, R., Weymouth, G. D., and Yue, D. K., "Hydrodynamic object recognition using pressure sensing," in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences.* The Royal Society, 2010, p. rspa20100095.

Boulogne, L. H., Wolf, B. J., Wiering, M. A., and van Netten, S. M., "Performance of neural networks for localizing moving objects with an artificial lateral line," *Bioinspiration & biomimetics*, vol. 12, no. 5, p. 056009, 2017.

Chen, J., Engel, J., Chen, N., Pandya, S., Coombs, S., and Liu, C., "Artificial lateral line and hydrodynamic object tracking," in *Micro Electro Mechanical Systems, 2006. MEMS 2006 Istanbul. 19th IEEE International Conference on.* IEEE, 2006, pp. 694–697.

Coombs, S., Braun, C. B., and Donovan, B., "The orienting response of lake michigan mottled sculpin is mediated by canal neuromasts," *Journal of Experimental Biology*, vol. 204, no. 2, pp. 337–348, 2001.

Coombs, S., Janssen, J., and Webb, J. F., "Diversity of lateral line systems: evolutionary and functional considerations," in *Sensory biology of aquatic animals.* Springer, 1988, pp. 553–593.

Coombs, S. and Van Netten, S., "The hydrodynamics and structural mechanics of the lateral line system," *Fish physiology*, vol. 23, pp. 103–139, 2005.

De Boor, C., *A practical guide to splines.* Springer-Verlag New York, 1978, vol. 27.

Dijkgraaf, S., "The functioning and significance of the lateral-line organs," *Biological Reviews*, vol. 38, no. 1, pp. 51–105, 1963. [Online]. Available: http://dx.doi.org/10.1111/j.1469-185X.1963.tb00654.x

Engelmann, J., Hanke, W., Mogdans, J., and Bleckmann, H., "Neurobiology: hydrodynamic stimuli and the fish lateral line," *Nature*, vol. 408, no. 6808, p. 51, 2000.

Fernandez, V. I., Maertens, A., Yaul, F. M., Dahl, J., Lang, J. H., and Triantafyllou, M. S., "Lateral-line-inspired sensor arrays for navigation and object identification," *Marine Technology Society Journal*, vol. 45, no. 4, pp. 130–146, 2011.

Fernandez, V. I., Hou, S. M., Hover, F. S., Lang, J. H., and Triantafyllou, M. S., "Development and application of distributed mems pressure sensor array for auv object avoidance," Massachusetts Institute of Technology. Sea Grant College Program, Tech. Rep., 2009.

Glaessgen, E. and Stargel, D., "The digital twin paradigm for future nasa and us air force vehicles," in *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, 2012, p. 1818.

Harashima, M., Ferrari, L. A., and Sankar, P., "Spline approximation using kalman filter state estimation," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, no. 5, pp. 421–424, 1997.

Hastie, T., Tibshirani, R., and Friedman, J. H., *The elements of statistical learning data mining, inference, and prediction*, 2nd ed. Springer, 2011, vol. 1.

Howe, B. M., Van Uffelen, L., Nosal, E.-M., and Carter, G. S., "Acoustic seagliders in philsea10: Preliminary results," in *OCEANS 2011*. IEEE, 2011, pp. 1–8.

Jauch, J., Bleimund, F., Rhode, S., and Gauterin, F., "Recursive b-spline approximation using the kalman filter," *Engineering Science and Technology, an International Journal*, vol. 20, no. 1, pp. 28–34, 2017.

Julier, S. J., "The scaled unscented transformation," in *American Control Conference, 2002. Proceedings of the 2002*, vol. 6. IEEE, 2002, pp. 4555–4559.

Kottapalli, A., Asadnia, M., Haghighi, R., Cloitre, A., y Alvarado, P. V., Miao, J., and Triantafyllou, M., "Mems sensors for assessing flow-related control of an underwater biomimetic robotic stingray," *Bioinspiration and Biomimetics*, vol. 10, p. 3, 2015.

Kottapalli, A., Asadnia, M., Miao, J., and Triantafyllou, M., "Artificial fish skin of self-powered micro-electromechanical systems hair cells for sensing hydrodynamic flow phenomena," *J. R. Soc. Interface*, vol. 12, pp. 1–14, 2015.

Maertens, A., "Touch at a distance : underwater object identification using pressure sensors," Thesis (S.M.), Massachusetts Institute of Technology, 2011.

Maertens, A. P., Dahl, J. M., and Triantafyllou, M. S., "Distributed pressure sensing to locate and identify obstacles," in 17^{th} *International Unmanned Untethered Submersible Technology (UUST 2011) Conference*, Portsmouth, NH, USA., aug 2011.

Maertens, A. P. and Triantafyllou, M. S., "The boundary layer instability of a gliding fish helps rather than prevents object identification," *Journal of Fluid Mechanics*, vol. 757, pp. 179–207, 2014.

Mathworks. "What is machine learning? 3 things you need to know." 2018. [Online]. Available: https://www.mathworks.com/discovery/machine-learning.html

Mivehchi, A., Dahl, J., and Licht, S., "Heaving and pitching oscillating foil propulsion in ground effect," *Journal of Fluids and Structures*, vol. 63, pp. 174 – 187, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0889974615300657

Partridge, B. L. and Pitcher, T. J., "The sensory basis of fish schools: Relative roles of lateral line and vision," *Journal of comparative physiology*, vol. 135, no. 4, pp. 315–325, Dec 1980. [Online]. Available: https://doi.org/10.1007/BF00657647

Press, W. H. and Teukolsky, S. A., "Savitzky-golay smoothing filters," *Computers in Physics*, vol. 4, no. 6, pp. 669–672, 1990.

Ramtilak, A., Joseph, A., Sivakumar, G., and Bhat, S., "Digital twin spark ignition for improved fuel economy and emissions on four stroke engines," SAE Technical Paper, Tech. Rep., 2005.

Renzi, D., Maniar, D., McNeill, S., Del Vecchio, C., *et al.*, "Developing a digital twin for floating production systems integrity management," in *OTC Brasil*. Offshore Technology Conference, 2017.

Roemmich, D., Johnson, G. C., Riser, S., Davis, R., Gilson, J., Owens, W. B., Garzoli, S. L., Schmid, C., and Ignaszewski, M., "The argo program: Observing the global ocean with profiling floats," *Oceanography*, vol. 22, no. 2, pp. 34–43, 2009.

Roman, C., Inglis, G., and Rutter, J., "Application of structured light imaging for high resolution mapping of underwater archaeological sites," in *OCEANS 2010 IEEE-Sydney*. IEEE, 2010, pp. 1–9.

Söderberg, R., Wärmefjord, K., Carlson, J. S., and Lindkvist, L., "Toward a digital twin for real-time geometry assurance in individualized production," *CIRP Annals*, vol. 66, no. 1, pp. 137–140, 2017.

Tao, F., Cheng, J., Qi, Q., Zhang, M., Zhang, H., and Sui, F., "Digital twin-driven product design, manufacturing and service with big data," *The International*

Journal of Advanced Manufacturing Technology, vol. 94, no. 9, pp. 3563–3576, Feb 2018. [Online]. Available: https://doi.org/10.1007/s00170-017-0233-1

Tuegel, E. J., Ingraffea, A. R., Eason, T. G., and Spottswood, S. M., "Reengineering aircraft structural life prediction using a digital twin," *International Journal of Aerospace Engineering*, vol. 2011, 2011.

Venturelli, R., Akanyeti, O., Visentin, F., Jeov, J., Chambers, L. D., Toming, G., Brown, J., Kruusmaa, M., Megill, W. M., and Fiorini, P., "Hydrodynamic pressure sensing with an artificial lateral line in steady and unsteady flows," *Bioinspiration & Biomimetics*, vol. 7, no. 3, p. 036004, 2012. [Online]. Available: http://stacks.iop.org/1748-3190/7/i=3/a=036004

Von Campenhausen, C., Riess, I., and Weissert, R., "Detection of stationary objects by the blind cave fishanoptichthys jordani (characidae)," *Journal of comparative physiology*, vol. 143, no. 3, pp. 369–374, 1981.

Wan, E. A. and Van Der Merwe, R., "The unscented kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000.* Ieee, 2000, pp. 153–158.

Webb, J. F. and Ramsay, J. B., "New interpretation of the 3-d configuration of lateral line scales and the lateral line canal contained within them," *Copeia*, vol. 105, no. 2, pp. 339–347, 2017.

Webb, J., "Lateral line structure," *Encyclopedia of fish physiology: From genome to environment*, vol. 1, pp. 336–346, 2011.

Weymouth, G. D., "Lily pad: Towards real-time interactive computational fluid dynamics," *Numerical Towing Tank Symposium*, vol. 18, oct 2015.

Weymouth, G. D. and Yue, D. K.-P., "Boundary data immersion method for cartesian-grid simulations of fluid-body interaction problems," *Journal of Computational Physics*, vol. 230, no. 16, pp. 6233–6247, jul 2011.

Windsor, S. P. and McHenry, M. J., "The influence of viscous hydrodynamics on the fish lateral-line system," *Integrative and comparative biology*, vol. 49, no. 6, pp. 691–701, 2009.

Yang, Y., Chen, J., Engel, J., Pandya, S., Chen, N., Tucker, C., Coombs, S., Jones, D. L., and Liu, C., "Distant touch hydrodynamic imaging with an artificial lateral line," *Proceedings of the National Academy of Sciences*, vol. 103, no. 50, pp. 18 891–18 895, 2006. [Online]. Available: http://www.pnas.org/content/103/50/18891