

AUDIO BEAMFORMING USING MACHINE LEARNING AND WIND TUNNEL
MODIFICATIONS

A Project

Presented to the

Faculty of

California State University, Fullerton

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Mechanical Engineering

By

Andrew Bartels

Project Committee Approval:

Salvador Mayoral, Department of Mechanical Engineering
Hope Weiss, Department of Mechanical Engineering
Darren Banks, Department of Mechanical Engineering

Summer, 2019

ABSTRACT

A low speed wind tunnel is a critical tool for engineers to test and validate new theories. As the history of wind tunnels have progressed, the need for more complex test setups were needed in order to study new topics such as aeroacoustics. The purpose of this project is to support future research topics of the wind tunnel facility at California State University, Fullerton. The three main goals of this project are: the foundation of a modular wind tunnel test section while further increasing the anechoic treatment of the test section, generate a prototype three-axis traverse to later collect quantitative measurements in the flow field, and finally, create a machine learning model capable of reproducing a beamformed acoustic signal of 7 microphones using only 2 microphones as inputs. The modification of the wind tunnel came in the form of numerous renovative processes, mainly the placement of a permanent dowel pin in the center of the test section (axially) creating a standard datum to measure from in the future. The prototype design of the three-dimensional traverse was largely successful, with all parts being installed as planned. The machine learning research showed promising results, with 2 beamformed microphones fed into the Long Short-Term Memory (LSTM) neural network, the reproduction of a sound pressure level signal with the noise reduction of 7 microphones being produced. Although a proof of concept, further investigations into machine learning could lead to a more reliable data processing tool in the Wind tunnel lab for further research.

TABLE OF CONTENTS

ABSTRACT.....	ii
LIST OF FIGURES	vi
ACKNOWLEDGMENTS	ix
Chapter	
1. INTRODUCTION	1
Wind tunnel.....	1
Aeroacoustics.....	6
Machine learning	10
2. BACKGROUND	12
3. EXPERIMENTAL SETUP & METHODS	19
Test setup and overview	19
Test section renovation	23
Traverse	29
4. LSTM METHODS AND RESULTS	37
Preprocessing	37
Network Architecture and Results.....	42
5. CONCLUSION.....	48
Test section renovation	23
Traverse	29
LSTM.....	29
REFERENCES	57

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1. Wind Tunnel	2
2. Traditional setup with stationary top and bottom panels	2
3. Old anechoic test setup with two axis traverses (1st revision)	3
4. Rolling road CAD model	3
5. Complete view of the anechoic chamber	5
6. Full view of the old anechoic top and bottom panel.....	5
7. CAD model cross section of datums (circled in red) and chamfers (teal)	7
8. Boeing's 787 engine and nacelle chevrons	9
9. Landing gear toboggan fairings	9
10. Flow past an aerodynamic body with an attached noise source.	10
11. Visual hierarchy of artificial intelligence	13
12. Different subsections of machine learning	14
13. Example of one perceptron	15
14. Simple explanation of neural networks.....	15
15. Visual representation of a CNN.....	16
16. A forward and backward pass through a neuron	16
17. Unfolded RNN	17
18. LSTM network architecture.....	18
19. Top view of anechoic chambers and test section.....	19

20.	Top and bottom anechoic cavities.....	20
21.	Extruded aluminum uniformly tensioned Kevlar panel next to test section.....	21
22.	Kevlar panel mounting in progress, latches shown in red	22
23.	Linear microphone array.....	23
24.	3 of 4 chambers installed with airfoil and traverse	23
25.	Grinding MIG welded holes flush with frame	25
26.	Wind tunnel section view with dowel pins and critical dimensions shown	26
27.	Datum A schematic	27
28.	Jig used to drill dowel pin holes with hardened steel collect in center hole	28
29.	Installed dowel pin with chamfers installed.....	28
30.	Anechoic test section with completed traverse installed	29
31.	Two axis traverse (first revision)	30
32.	Two axis traverse (second revision)	31
33.	Cal Poly low speed wind tunnel.....	32
34.	Cal Poly low speed 3-axis probe traverse	32
35.	Iterative design process used	33
36.	Traverse CAD model with labeled axes	34
37.	ZX Gantry system	35
38.	Static X axis assembly components	36
39.	Simple machine learning pipeline.....	38
40.	Example output of the LSTM preprocessing MATLAB script	39
41.	Mel spectrogram function MATLAB	40
42.	Mel filter bank design	41

43.	Mel-frequency Spectrogram	41
44.	RMSE LSTM network comparison graph.....	43
45.	Validation loss LSTM network comparison graph.....	44
46.	A simple LSTM network architecture	44
47.	A Deep C-BiLSTM network architecture.....	45
48.	Prediction 28 of a simple LSTM.....	46
49.	Prediction 122 of a simple LSTM.....	47

ACKNOWLEDGMENTS

I would like to thank my project advisor, Dr. Salvador Mayoral, Assistant Professor of the Mechanical Engineering Department at California State University, Fullerton. He provided me not only the opportunity to work on an exciting design problem supported and encouraged my growing interests in machine learning and aeroacoustics. He consistently offered insight and support whenever problems arose. I would also like to thank my thesis committee: Dr. Hope Weiss and Dr. Darren Banks for their support and assistance. Lastly, I would like to thank the supporting member(s) of the center for collaborative research and prototype development for their continued support and engineering experience, I could not have had success and results without them.

CHAPTER 1

INTRODUCTION

Wind Tunnel

Wind tunnels are an important tool for aeronautical research and development [1].

Study topics have expanded for wind tunnels from the “traditional” era of studying lift and drag phenomena, to current day, where the interactions between bodies of study and acoustics have been of interest [2,3] along with ground effect testing on aerodynamic bodies [36].

A traditional low speed wind tunnel as shown in Figure 1 has a fan (3) that causes a steady flow of air starting in the settling chamber (1) and into the test section (2) where the object of study and test equipment reside, before passing through the diffuser (between 2 and 3) and out the back.

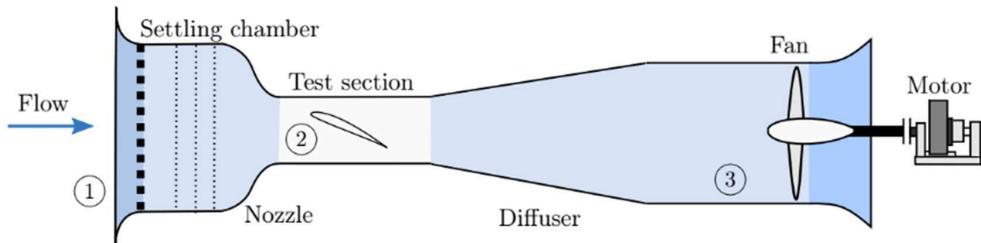


Figure 1. Wind Tunnel [4].

The low speed wind tunnel currently residing at California State University, Fullerton’s wind tunnel lab was originally installed with a “standard” test section of wood inside a steel framework sometime in the 1970s (estimated). Since then, the test section has been largely upgraded with up to date technology and a modern data acquisition

system (DAQ). Different setups for the wind tunnel include a traditional setup with 4 stationary walls (Figure 2), an anechoically treated setup (Figure 3), and in the future a “rolling road” configuration (Figure 4) that is currently in development. All testing setups currently require a traverse which is used for quantitative measurements of the flow field.

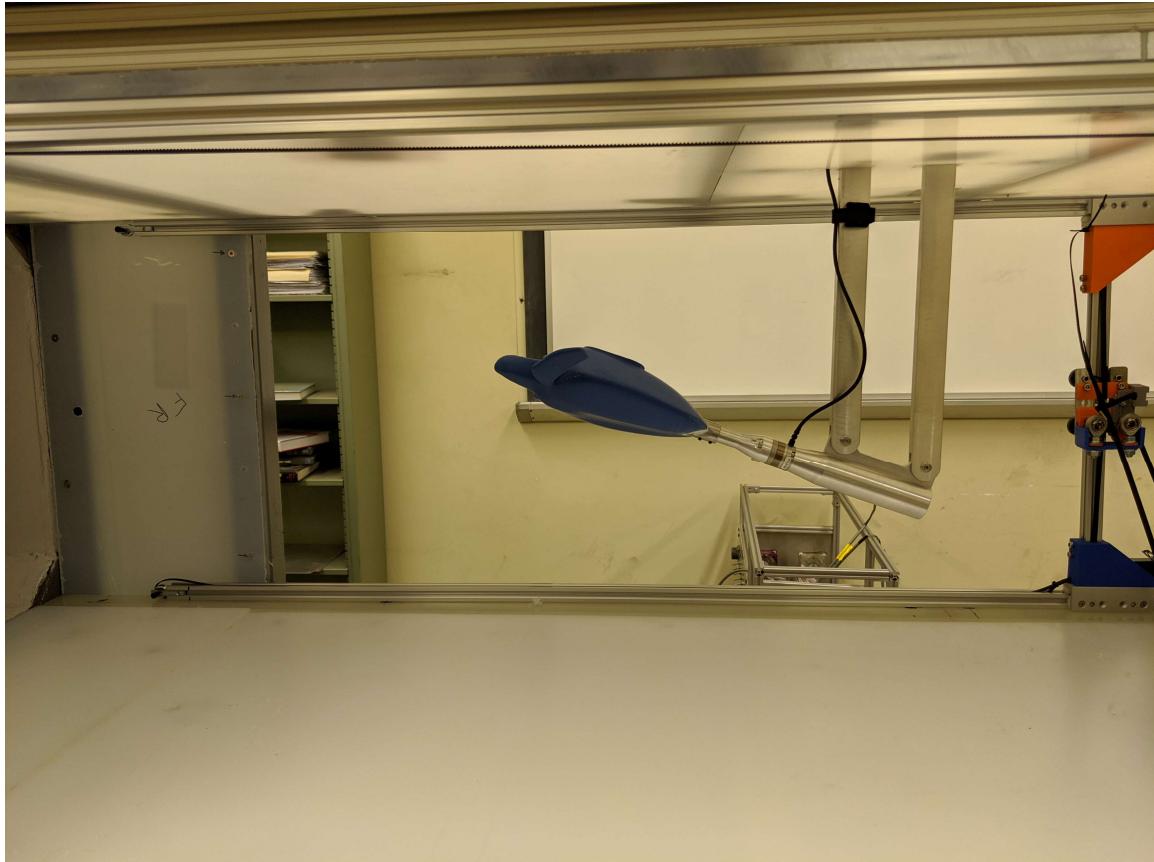


Figure 2. Traditional setup with stationary top and bottom panels

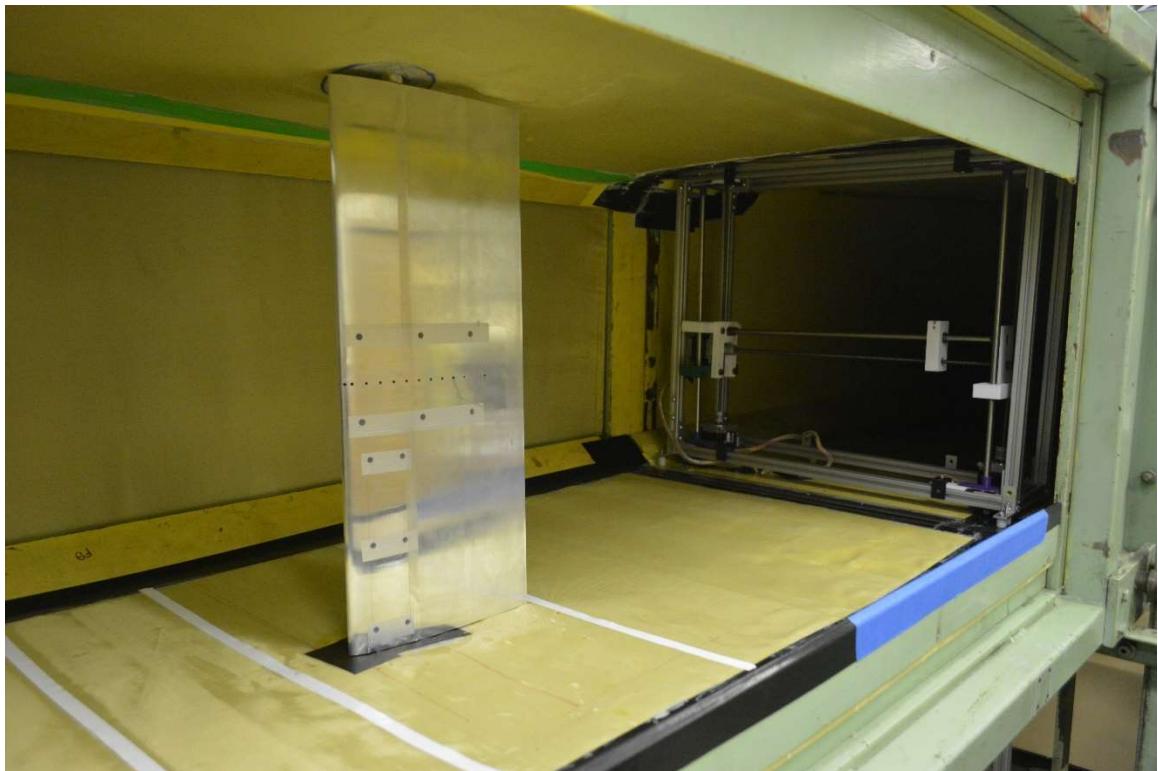


Figure 3. Old anechoic test setup with two axis traverses (1st revision) [3].

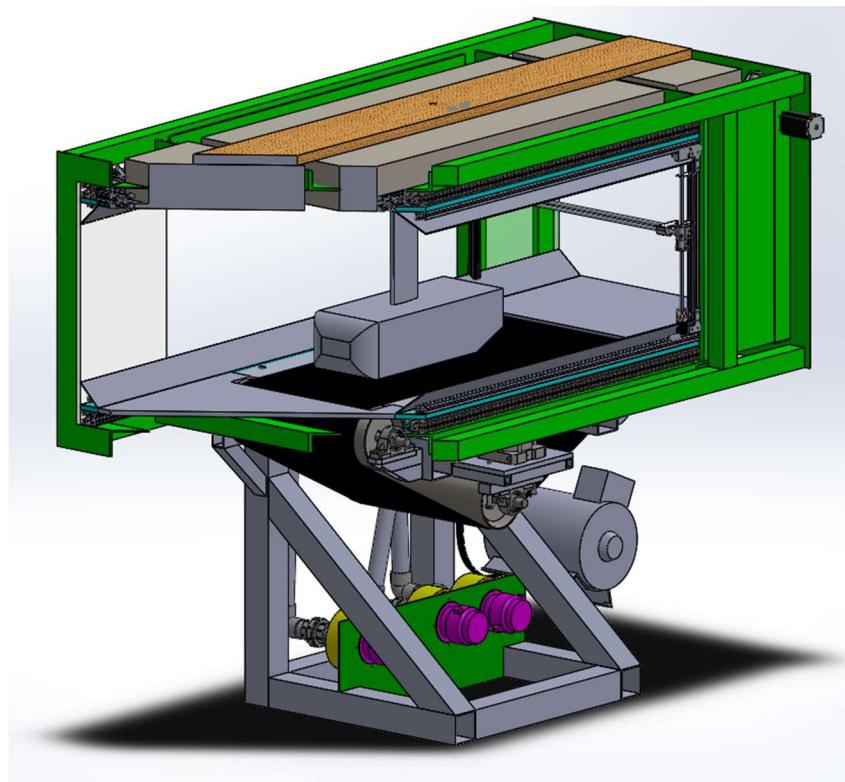


Figure 4. Rolling road CAD model

These different setups in the past often required custom fit panels to accommodate the 1.5" width difference in the horizontal direction from front to back (axially) of the test section. These custom fit panels were difficult to secure to the test section and difficult to change with the other configurations. The two-axis traverse would also not allow for any testing data to be collected for an object of study in the axial (third) dimension. Collectively, switching between test setups can range from a few hours to days, and required the traverse to be taken out of the test section if a different test section configuration was needed.

The anechoic wind tunnel configuration is based largely off Virginia Tech's 1.8 m x 1.8 m Stability Tunnel [12,13]. Previous tests [3,4] used SONEX polyurethane acoustical foam to treat the walls of the anechoic chambers as seen in Figure 5. for the left and right sides, while the top and bottom panels (Figure 6) were uniformly tensioned Kevlar screens with polyurethane acoustical foam directly behind the Kevlar. The airfoil or object of study was then mounted horizontally in the center of the test section using the white ABS mount in Figure 7. The objective of all these modifications is to mimic a free field environment inside the wind tunnel, where all reflections and external noises are absorbed except the object(s) of study.



Figure 5. Complete view of the anechoic chamber [3]



Figure 6. Full view of the old anechoic top and bottom panel

In developing the newer testing facility, numerous, very arduous, subitems were required consuming roughly 1,500 work hours in total. In pursuit of a test section that could better align with the requirements of the lab, the first item was stripping the test section down to the frame and MIG welding all the numerous holes closed in the test section accumulated over its life (roughly 50 in all). After angle grinding all the welds flush with the frame, dowel pins were permanently mounted using a jig to precision drill and ream the holes, press fitting the dowel pins in place. The dowel pins served as a datum for any future needed measurements and an aligning feature for all panels placed in the test section as seen in Figure 5. Chamfer pieces would be mounted in the test section (shown in teal, Figure 7) on top of permanently mounted 80/20 aluminum profiles. The chamfers were placed in all four corners of the test section which would be flush with any extruded aluminum profile panels placed on the top and bottom of the wind tunnel test section. Next, the top and bottom Kevlar panels were 3D modeled and created out of 1/2" x 2" extruded aluminum and uniformly tensioned using thin steel strips and sliding nuts and bolts. The anechoic top and bottom panels of previous designs were replaced with anechoically treated cavities akin to the side chambers used previously by Dousti and Khader [2, 3].

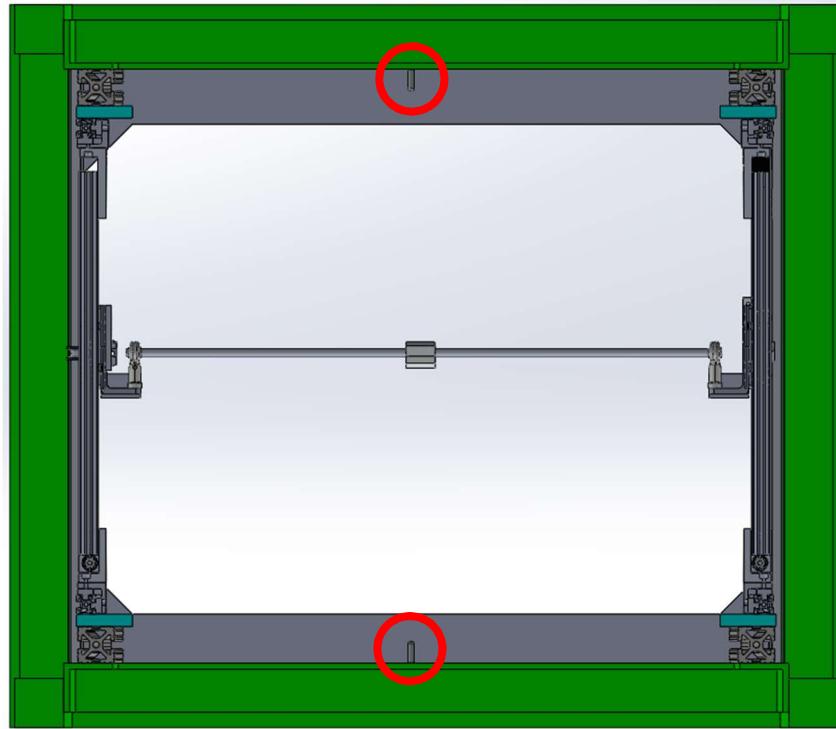


Figure 7. CAD model cross section of datums (circled in red) and chamfers (teal)

In parallel to the wind tunnel modifications, the three-axis traverse (simply referenced as “the traverse”) was designed in 3D modeling software [7]. The traverse concept was modeled after modern fused deposition modeling (FDM) printing frames. All parts used in the manufacturing process were mostly off the shelf general 3D printer parts or printed on two FDM printers available at the time. The iterative design process [9] was used to reach a more final proof of concept, with critical design points being the durability and rigidity of the traverse carriages while having the most minimal blockage in the test section. This new traverse was designed to have a very low profile in the test section while being able to actuate in the X, Y, and Z axes to collect turbulence data virtually anywhere in the test section.

Aeroacoustics

NASA's N+2 and N+3 requirements are asking for a -42dB and -71dB noise reduction for all landing and takeoff of commercial traffic by 2020 and 2025 respectively [9]. This along with other research by Papamoschou et al. [10] have fueled an interest in the study of aircraft noise reduction specifically in the wind tunnel lab at Cal State Fullerton. Aircraft noise is generally grouped into two main categories: airframe noise, and engine noise. Airframe noise is produced by all non-propulsive parts which mainly occur during landing and takeoff phases of flight. Components include the trailing and leading edges of wings, flap side edges, and slat/flap coves. Alternatively, engine noise can be produced from fan noise, or flow over turbine fan blades, or jet noise which is produced by the mixing of turbulent eddies of shearing flow and high velocity jets.

Noise reduction items as seen on the Boeing 787's engine chevrons and landing gear fairings (Figure 8 and Figure 9) have been topics of study for some time and address larger components of both airframe and engine noise [11]. These areas have covered two large issues for aircraft noise reduction but have left crucial remaining questions such as the noise reduction over the wake of the aircraft body itself and the trailing edges of the wing.

Investigation of these aft-emitting noise source by a momentum wake were further investigated by Khader and Dousti [3] in addition to trailing edge devices such as serrations somewhat akin to Boeing's chevrons on the trailing edge of an airfoil [2].

Figure 10 illustrates the summation of what aeroacoustic phenomena is being studied.



Figure 8. Boeing's 787 engine and nacelle chevrons [11]



Figure 9. Landing gear toboggan fairings [11]

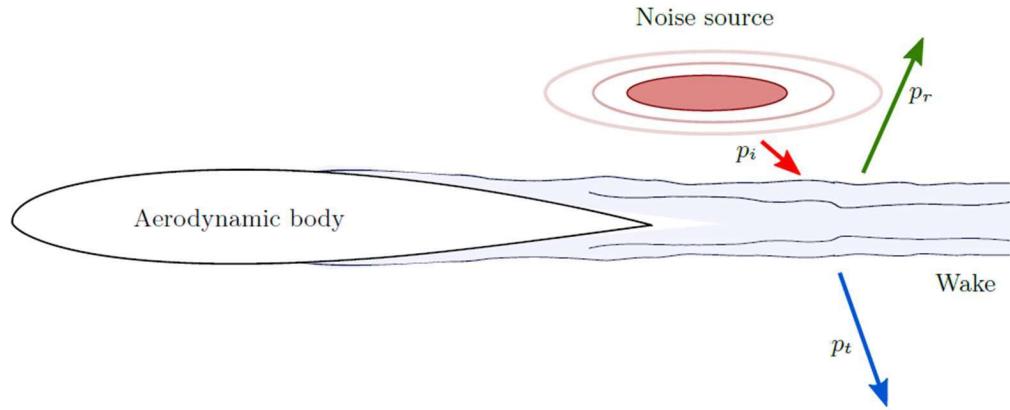


Figure 10. Flow past an aerodynamic body with an attached noise source. [3]

In Figure 10, P_t is the sound propagation through the wake is compared to the noise source, or the incident pressure wave (P_i) which is generated and known, and therefore the reflected pressure wave (P_r) can be found. In summation P_t is the data directly being observed by the microphone array and the noise reduction can be found from this. In this research, equation 1 elaborates the fundamental equation that is to be investigated. The sound pressure level (SPL, dB) of the microphones are processed from the background noise ($SPL_{background}$) with the noise source turned off. This cleaned SPL is then studied to see if there was a reduction in overall noise level through the wake of the airfoil.

$$SPL_{cleaned} = 10 \log_{10} \left[10^{\left(\frac{SPL_{uncleaned}}{10} \right)} - 10^{\left(\frac{SPL_{background}}{10} \right)} \right] \quad (1)$$

Machine Learning

The main goal of this project was to replicate the noise transmission loss test [3,4] and feed this new data into a machine learning model to predict the same noise suppression effects of 7 microphones using just 2 microphones. An LSTM network was the machine learning algorithm that was used. An application of this would be to use only

2 microphones, using the other 5 for other places in the wind tunnel, essentially enabling cheaply produced software to replace hardware in specific settings. In the wind tunnel lab, machine learning was not a component of research, so this research was the first of its kind.

The purpose of this research is supportive in nature to the overarching future research topics of the wind tunnel lab. This paper had three main goals to be accomplished and is documentation for a graduate research project at Cal State Fullerton. Although the topics are tangentially related, they are all in concert with furthering the development of the wind tunnel facility. Among the three goals to accomplish, the first was to generate a more modular test section that can better accommodate the changing of the 3 major setup: a standard wind tunnel, anechoic, and rolling road setup. A subtask of the first goal was to increase the anechoic treatment of the test section by replacing the top and bottom anechoic panels with removable cavities. The second goal was to produce a working concept of a three-axis traverse that is capable of being left in the test section for all configurations. In parallel with the traverse prototype was the last goal, development of a machine learning code base that is fed acoustic data to model sequence to sequence reproduction of a cleaned beamformed signal with all background noise removed. In order to fully validate the machine learning model, a separate set of acoustic data was required. This validation test set of data essentially required the completion of the first two objectives in order to reliably assess the performance of the machine learning model.

CHAPTER 2

BACKGROUND

Chapter 2 presents all needed background information of the project regarding any LSTM background information regarding a sequence-to-sequence machine learning model. Since it is important to understand LSTM models, it is necessary to start with a simple overview of all the components and brief history of machine learning before LSTM models are explained since they are a more complex architecture.

Machine learning and Artificial Intelligence (AI) are often topics overused by numerous industries as a remedy to all problems [14]. The terms “machine learning” and “artificial intelligence” are often terms used interchangeably but doing so creates imprecision and ambiguity [14]. Firstly, in explaining LSTM neural networks used in this research paper, a few items need to be clearly defined first. It is useful to make a distinction of what machine learning is and is not. Figure 11 is a visual overview of the subject and subsects of each topic.

AI is a program that can adapt and reason with new information that it has been given, where machine learning is often thought as a subset of AI. Machine learning is an algorithm or grouping of algorithms, and whose performance improves as models are exposed to more data. Another useful definition is that a machine learning algorithm is often referred to as an algorithm that hasn’t been explicitly programmed [30]. This algorithm is given data, asked to categorize or produce and output, and figure out the in between as a kernelized mapping of the data. Furthermore, Figure 12 is another visual

aide that helps define the broad scope of machine learning, and the grouping of algorithms the term encompasses. Nested inside of the term machine learning are two major hierarchical branches of supervised and unsupervised learning. These terms are simply used to separate algorithms that require the data to be labelled, and thereby giving the optimization algorithm an “answer key” to observe if a prediction was accurate or not. Often it is implied that machine learning is referring to neural networks, which as seen in Figure 12 can be used for classification, regression, or clustering objectives. Figure A3 of Appendix A can help supplement a few applications of the different types of machine learning.

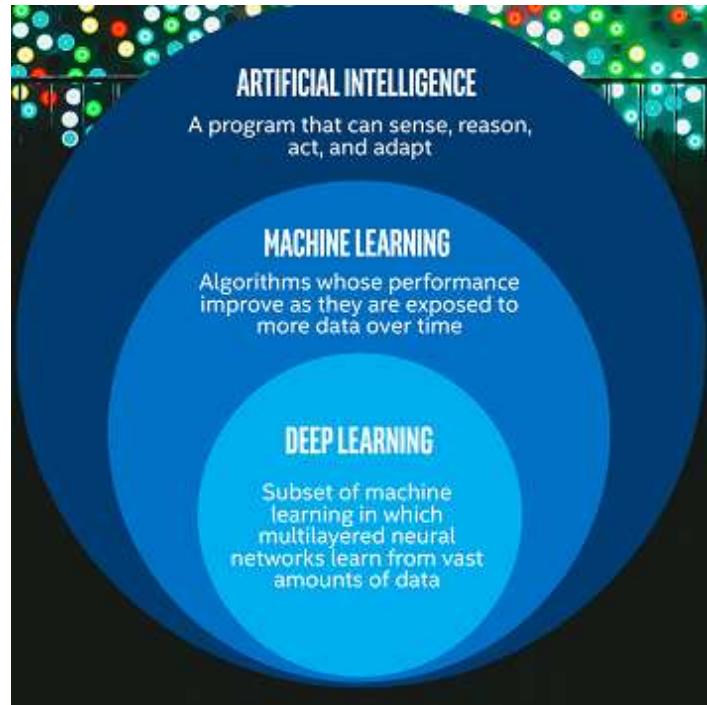


Figure 11. Visual hierarchy of artificial intelligence

For this research, supervised learning was used, explicitly a neural network for time series data (LSTM), which is a type of convolutional neural network (CNN) [15]. Before explaining a LSTM network, it is often best to start from the simplest building

block of these networks. The simplest building block of a convolutional neural network is a perceptron was generated by Rosenblatt in 1958 [28]. Figure 13 gives a simple graphic of a perceptron with an error calculation. This building block takes an input with randomly initialized weights and sums them into a function input, and is then pushed through a non-linear activation function, that is supposed to “fire” if the desired output is reached. If the desired output is reached incorrectly, an error is computed, and the weights are adjusted by an optimization function that determines if this coefficient is moved up or down in magnitude (Figure 16). The nonlinearity of the activation function is a notable detail, as this introduction of a nonlinearity allows a model (or neural network) to map response variables that varies non-linearly with respect to the inputs.

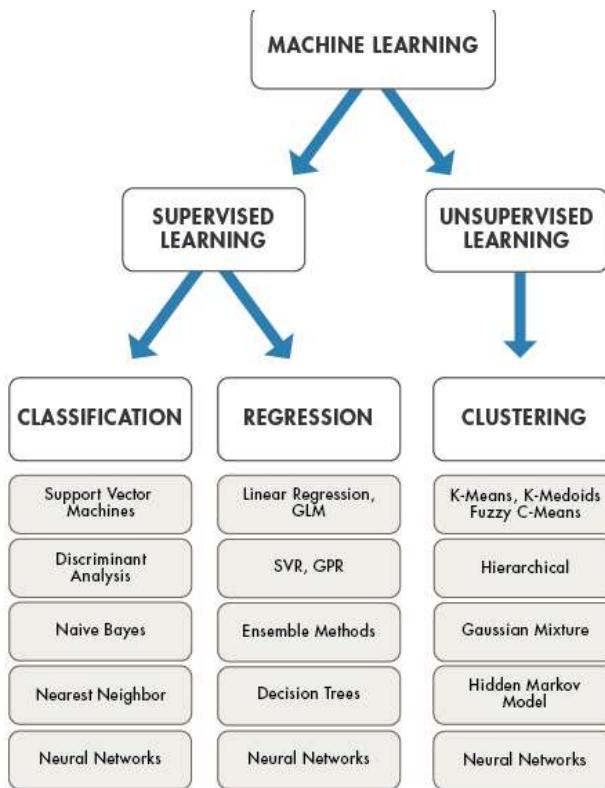


Figure 12. Different subsections of machine learning [15]

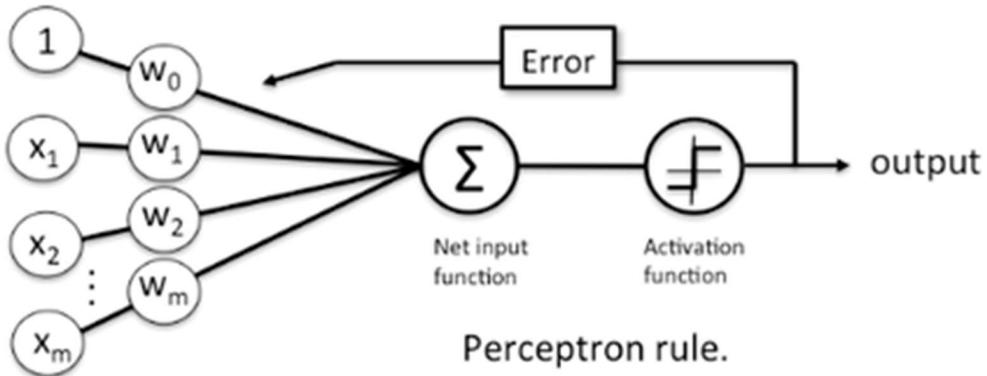


Figure 13. Example of one perceptron [28]

Figures 14 is a general representation of a more modern neural network where many perceptron are strung together into what is generally referenced as the hidden layers of a network. Figures 15 can help best represent how a CNN works. This additional step of convolution is essentially a kernelization of the input data (generally an image) along with a nonlinear activation function

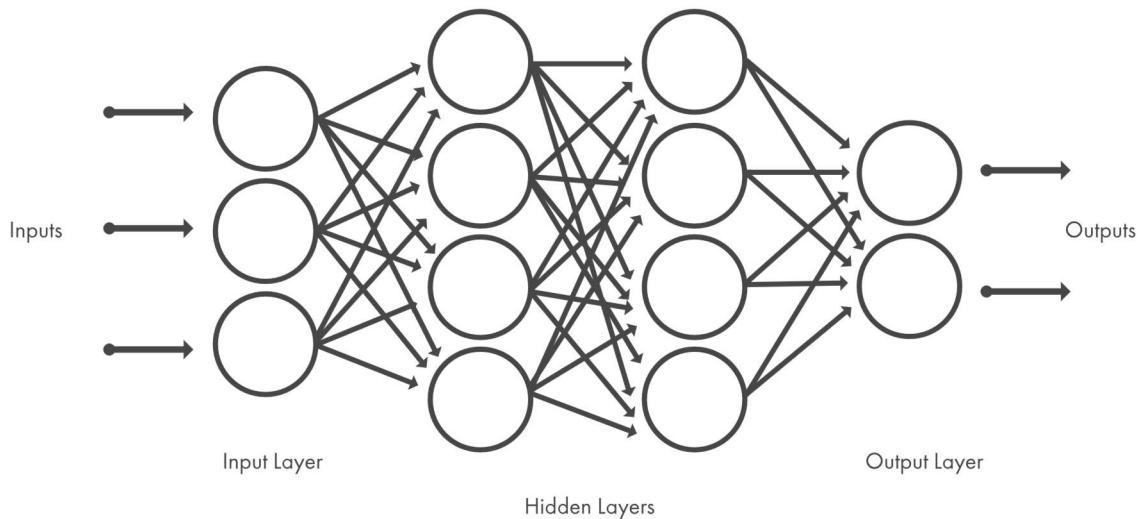


Figure 14. Simple explanation of neural networks [15]

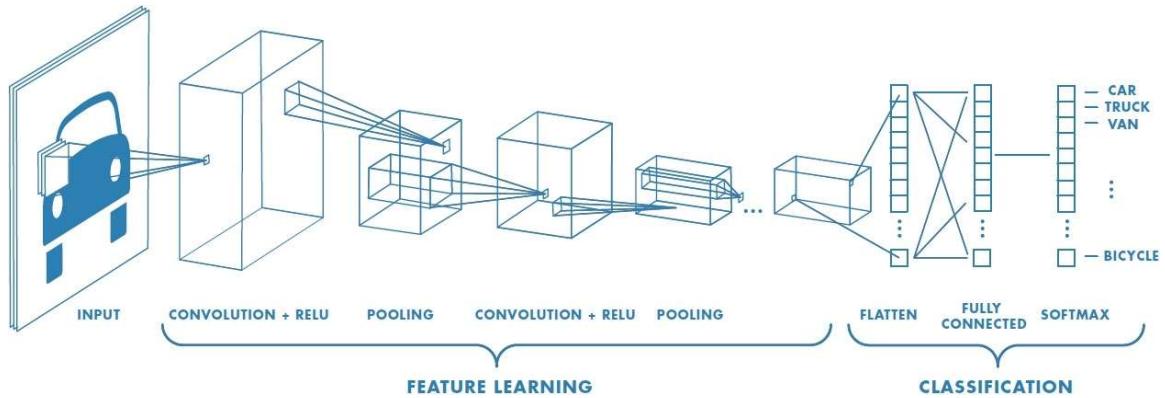


Figure 15. Visual representation of a CNN [15]

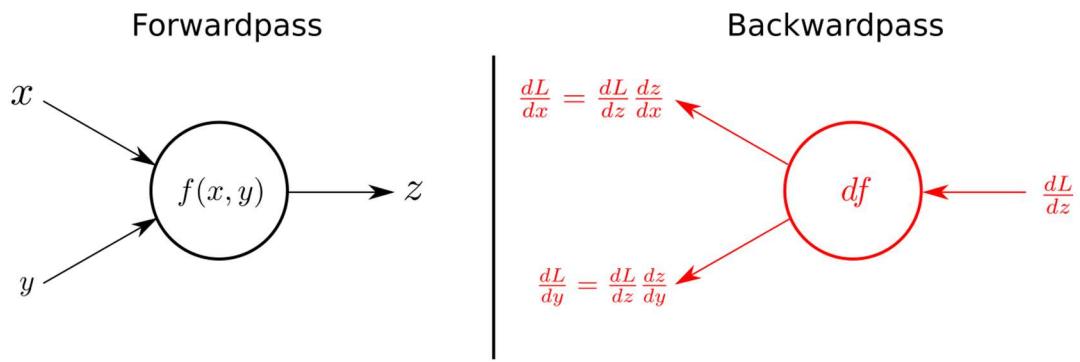


Figure 16. A forward and backward pass through a neuron. [18,19]

Simply put, a neural network is a higher order representation of data that is fed into the network, and features are extracted and distilled in a matrix of weights. These models are simply many optimized weights that when training is very computationally intense, but highly parallelizable using a Graphics Processing Unit (GPU) [16]. Once the neural network is trained, predictions on new inputs are very computationally light. However, the main issue for this research with CNN or “Vanilla” neural networks, is that there is no mechanism for representing any form of series in time.

Recurrent neural networks (RNN, Figure 17) are CNNs that store model states for each time step. However, in application RNNs encounter what is often referred to as the vanishing gradient problem as best stated by Hochreiter: “Recurrent nets are in principle

capable to store past inputs to produce the currently desired output. Because of this property recurrent nets are used in time series prediction and process control. Practical applications involve temporal dependencies spanning many time steps, e.g. between relevant inputs and desired outputs. In this case, however, gradient based learning methods take too much time. The extremely increased learning time arises because the error vanishes as it gets propagated back.” [17] This vanishing gradient problem issue occurs with long time (or in the case of this research, spectral sequences) sequences such as the Sound Pressure Level (SPL) signals that are used in this research. In such cases an LSTM network architecture is often implemented. As seen in Figure 18, an LSTM network architecture consists of a memory cell, input and output gates, a forget gate, and the hidden states which are the components of an RNN.

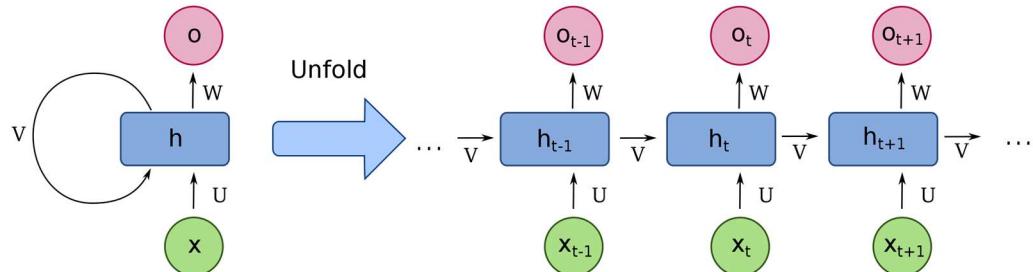


Figure 17. Unfolded RNN [19]

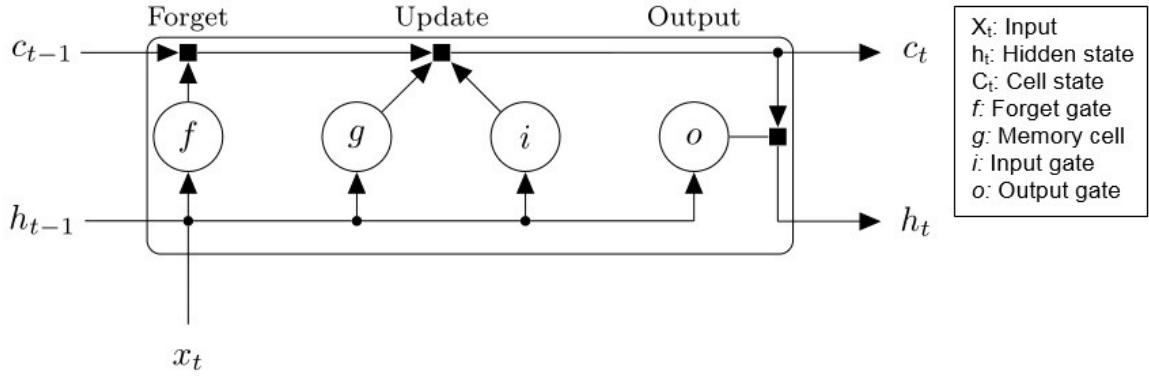


Figure 18: LSTM network architecture [20]

Before this research, the raw data captured from the array of 7 microphones were imported into MATLAB [23] and the separate microphone sensitivity profiles were imported and subtracted from a beamformed signal of the 7 microphones. Then a background noise file which is data collected without the loud speaker producing noise was then subtracted in the spectral domain. This “cleaned” beamformed signal was then plotted and analyzed for a transmission loss which was usually observed on an SPL versus frequency plot. The greater the difference between the noise generated by a loud speaker and the noise received by the microphones was white noise assumed to be dissipated by the turbulent wake of the airfoil in the test section.

CHAPTER 3

EXPERIMENTAL SETUP AND METHODS

Chapter 3 presents all experimental setup for conducting this study. Additionally, this chapter encompasses all the modifications that were done in order to update the test section and any preparations that could have a foreseeable benefit in future testing and research. All the design and methodology of the 3-axis traverse being installed in the test section are also included in this chapter.

Wind tunnel anechoic setup and overview

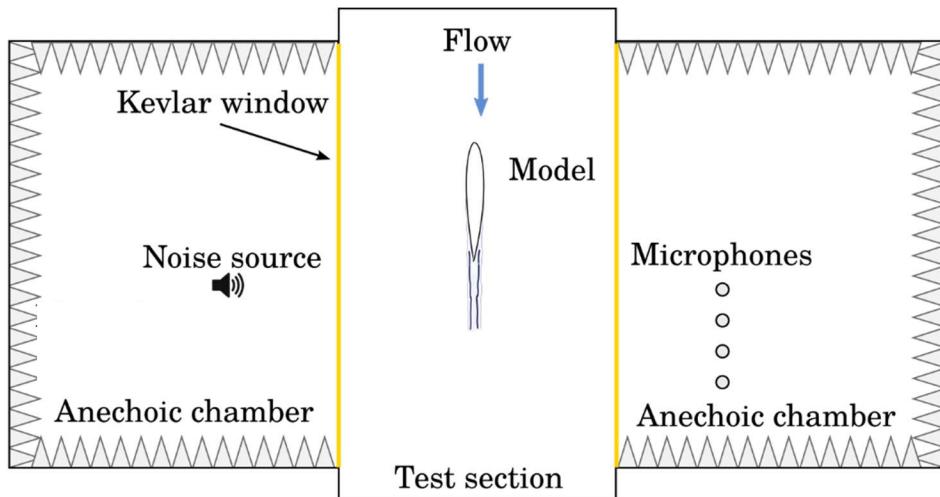


Figure 19. Top view of anechoic chambers and test section [4].

For the experimental setup, Figure 19 is a concise schematic of the validation data that was collected. In Figure 19 the left side anechoic chamber contains the noise source, a 3" diameter full range loudspeaker, attached to a HP E3630A waveform signal generator through which transmits white noise chambers. With the Kevlar screens covering both cavities, and a NACA 0012 airfoil in the center generating a wake. On the

right chamber, seven PCB microphones (model 130E21) in a linear array were moved in 10 positions from left to right collecting acoustic data at a 48,000-sampling rate with a National Instruments DAQ card.

An updated top and bottom sections were created with 8" polyurethane acoustical foam lining all the walls as seen in Figure 20. The Kevlar screens are mounted to separate frames and were created out of 1/2" x 2" extruded aluminum and uniformly tensioned using thin steel strips and sliding M5 nut and bolts (Figure 21)



Figure 20. Top and bottom anechoic cavities

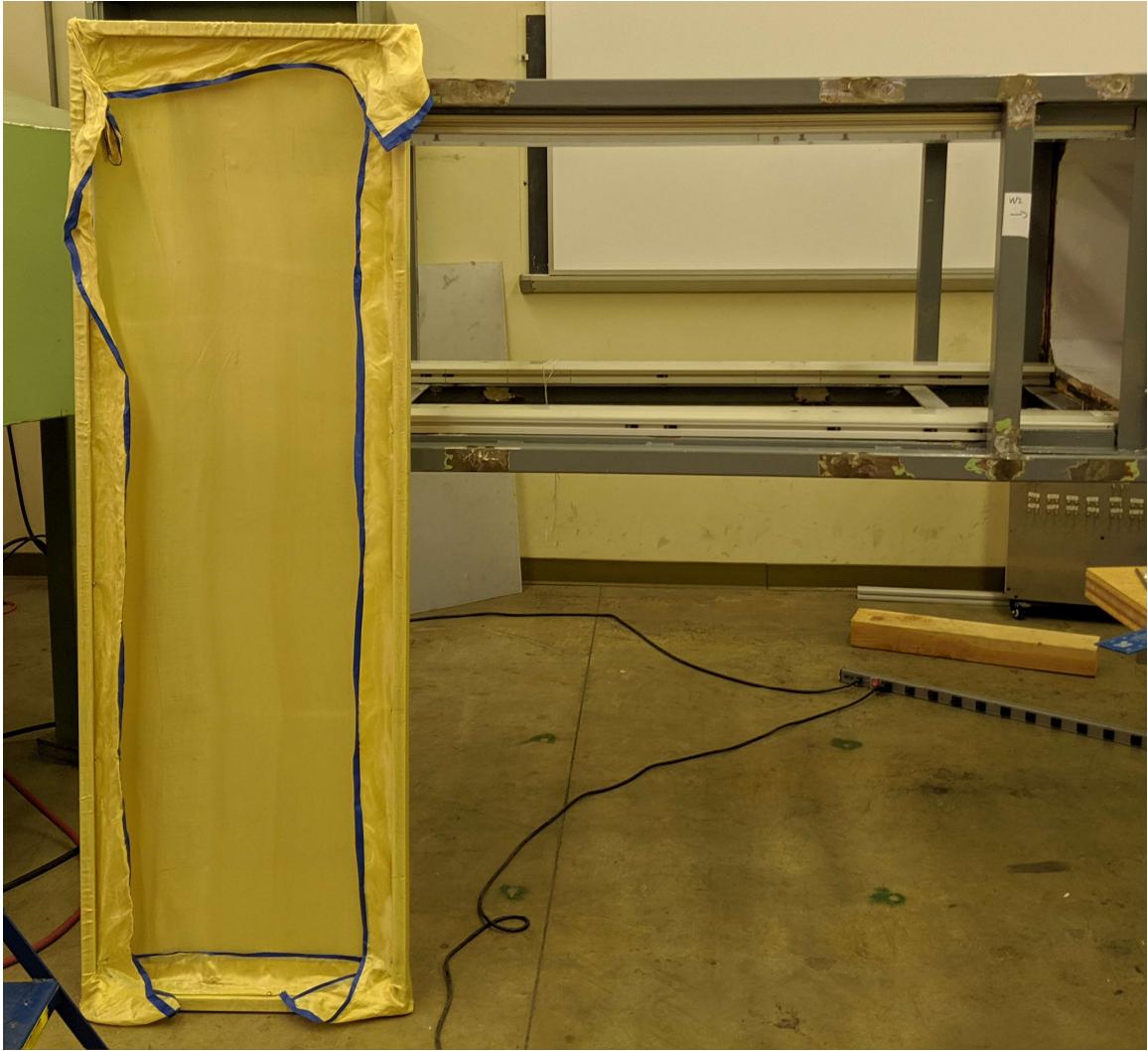


Figure 21. Extruded aluminum uniformly tensioned Kevlar panel next to test section

To mount the top and bottom Kevlar panels, jacks are stood on 4"x 4" wooden blocks, and raised in between the permanently mounted chamfers, where the latches are locked in place, and mounted to the frame on all four corners of the test section as seen in Figure 22. After mounting the Kevlar panels and cavities to the test section, the NACA 0012 airfoil is mounted into the test section. The microphones are then mounted to the railing system installed in the left side chamber (Figure 23). For validation set of data, the airfoil was simply there to generate a wake, and turbulence measurements were not taken during this test.

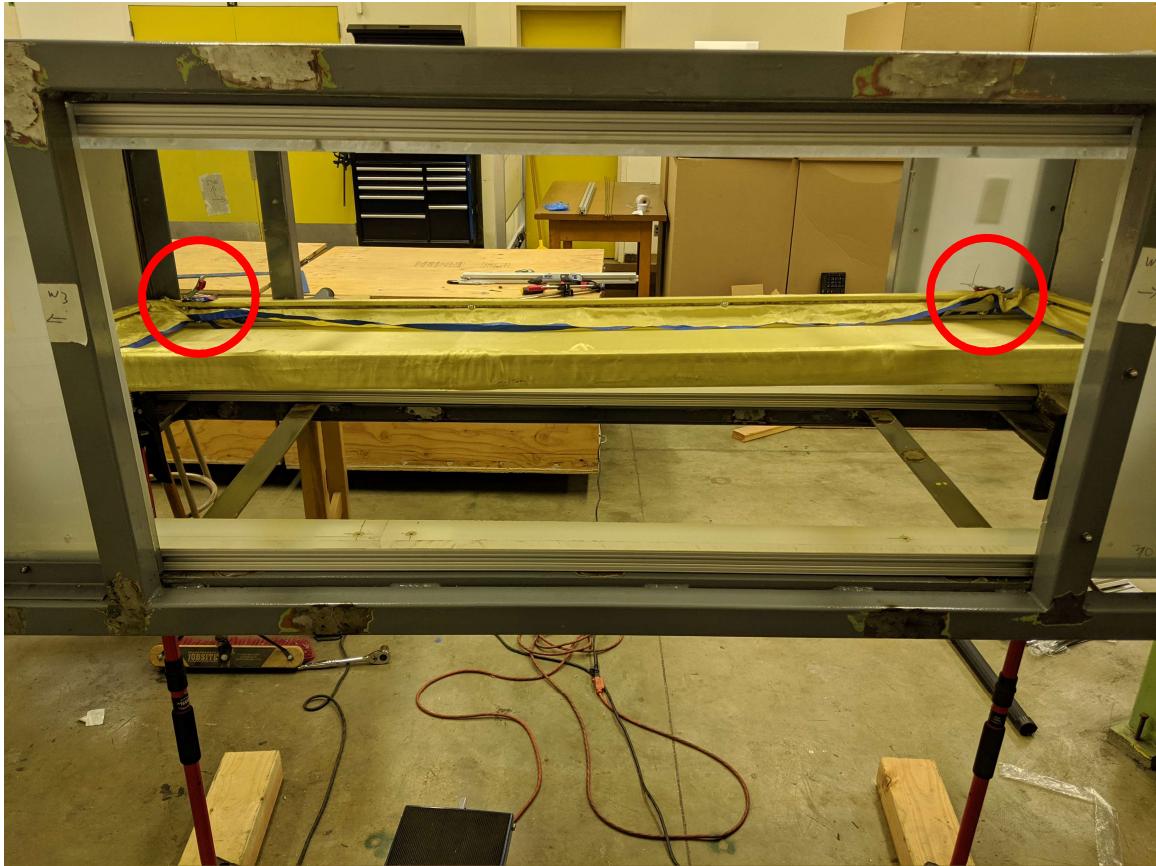


Figure 22. Kevlar panel mounting in progress, latches shown in red

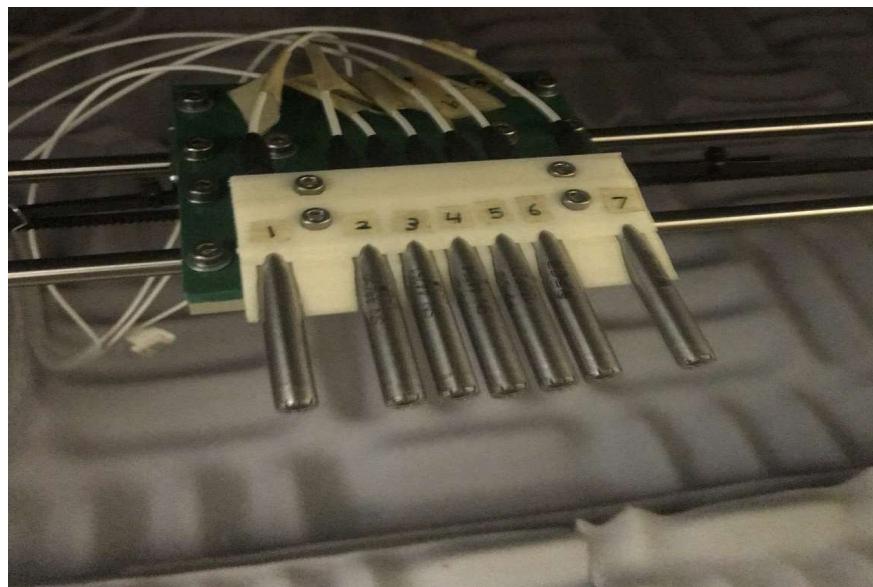


Figure 23. Linear microphone array



Figure 24. 3 of 4 chambers installed with airfoil and traverse

Figure 24 displays the completed test setup in its entirety. Testing included running the low speed wind tunnel from 5 m/s to 25 m/s at 7 different speeds. The movement of the microphone array was controlled using a LabVIEW VI program [22] while the DAQ system collected 48,000 samples per second and placed them in a digital audio tape (.dat) file for later processing using MATLAB [23].

Test section renovation

To standardize the wind tunnel test section, there were a handful of arduous but necessary tasks that are needed for further research to have defined mounting points with simple and easy to produce geometries for items such as the top and bottom panels of the test section. The ultimate product of this section of the research is to ensure different

configurations could be moved in and out of the test section with a drop in, latch, and test ideal.

Before any modifications could be installed, the wind tunnel had accumulated a decent number of holes in the test section over the years that could lead to an assortment of issues in the future. Roughly 50 holes in all were MIG welded shut and angle grinded flush with the frame as seen in Figure 25. Before the chamfer pieces were cut and mounted as a permanent feature that stays in the wind tunnel regardless of configuration, a conclusion was reached that a datum was needed in order to properly measure and verify chamfer alignment in the test section. The top and bottom halves of the test section are not perfectly symmetric to each other. In the past, this might not have been an issue, but if four separate chamfers were being place on top of 1.5" x 1.5" extruded aluminum, this could lead to two rectangular panels of the same dimensions fitting on the bottom, but not the top (or visa versa). Using Figure 26 for context, it is clear a datum will help constrain the critical dimension where top and bottom panels will fit.

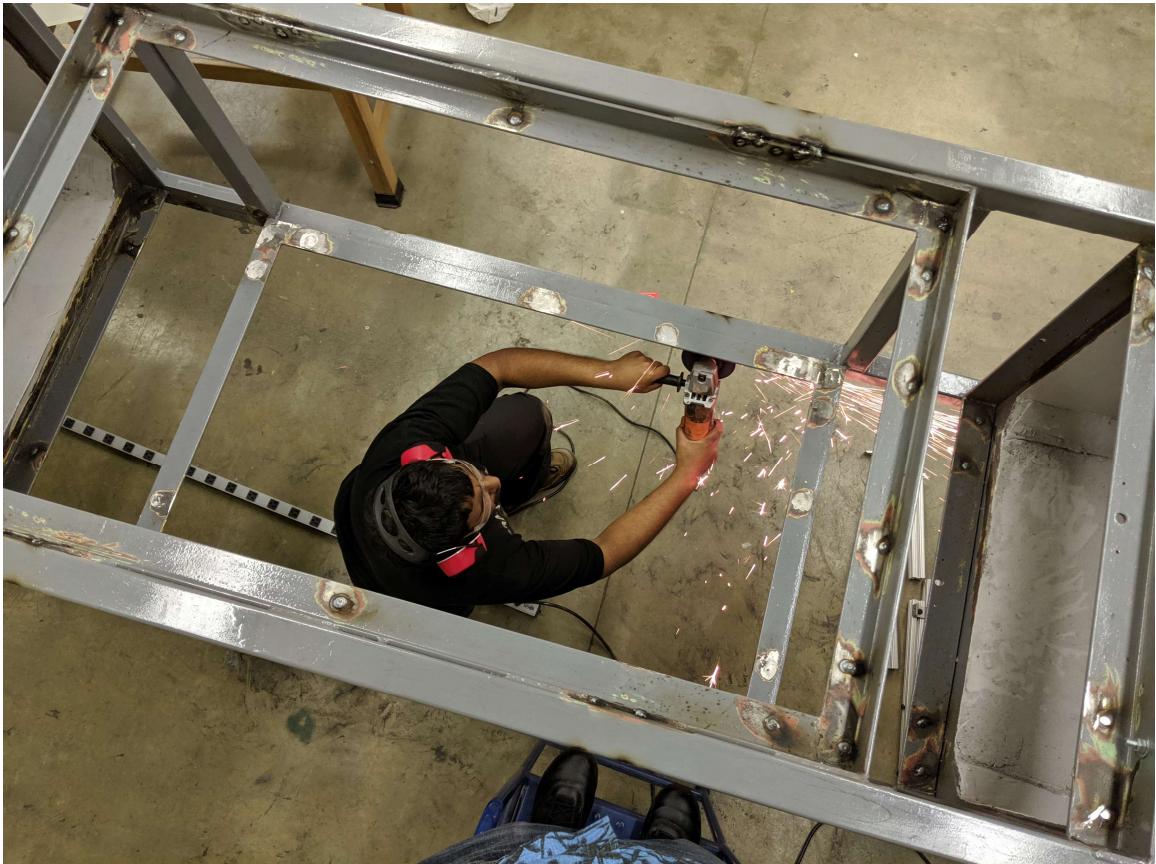


Figure 25. Grinding MIG welded holes flush with frame

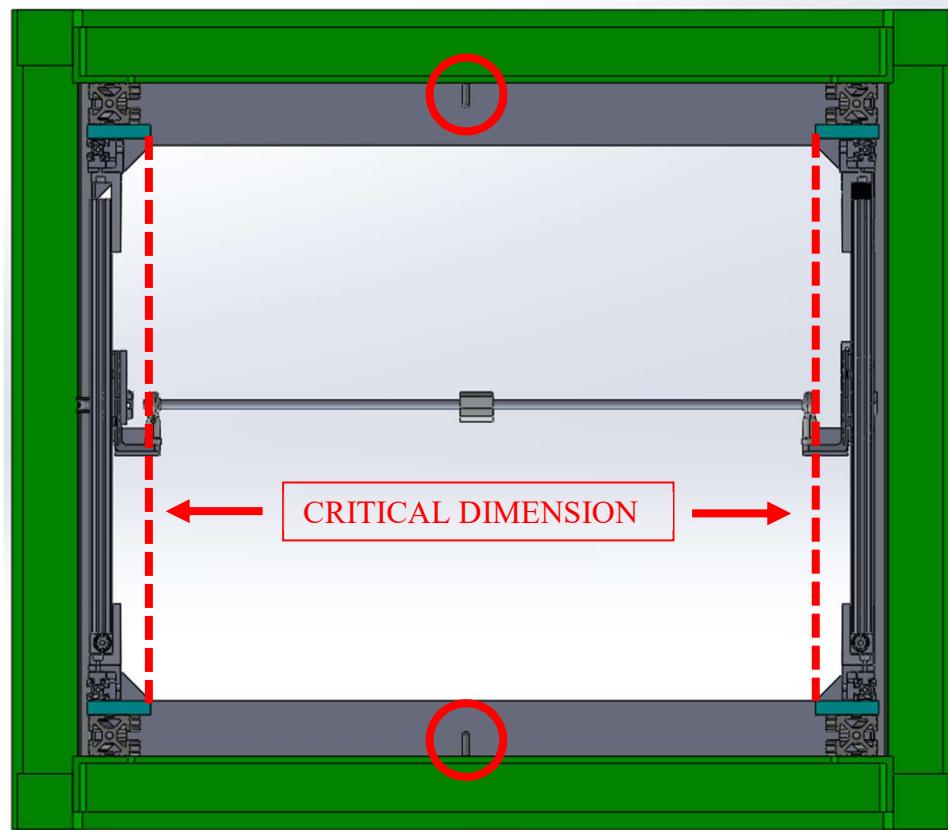


Figure 26. Wind tunnel section view with dowel pins and critical dimensions shown

This datum in the form of a .2501" press fit dowel pin is a crucial feature in the development of the test section. The 1.5" width difference in the test section can generate many issues for measurement and placement of objects being mounted into the test section. This dowel pin serves also as an aligning feature for any 1/2" x 2" extruded aluminum rectangular panels to be placed flush with the chamfers on the top and bottom of the wind tunnel. Any panels being placed in the test section will now require a 1/4" hole in the midpoint of the panel where the dowel will fit. Figure 27 displays the plane that is generated by the dowel pin features. The holes drilled into the test section flanges were precision drilled using a jig which was cut and drilled on a Bridgeport manual mill with digital readouts. The hardened steel collet displayed in the center of Figure 28, was

then pressed into the jig and mounted using clamps and shimmed onto the frame. A hand drill was then used to drill the holes to an undersized 1/4" and then reamed with a .2500" ream. This meticulous setup ensured that the dowel pins were correctly placed onto the datum "A" plane (Figure 27), while being perpendicular to the flanged the hole was drilled on. The pins were then carefully fit into the holes using a ball pin hammer. The final tolerance goal was an ANSI LN2 interference fit which is anywhere from 0" - .001" interference between the shaft and hole. This allows the pins to be placed permanently in the frame without fear of the pins being knocked out of their holes, while foregoing the need of an industrial pneumatic press to press fit them in. Figure 29 displays a dowel pin inside the test section with the chamfers installed

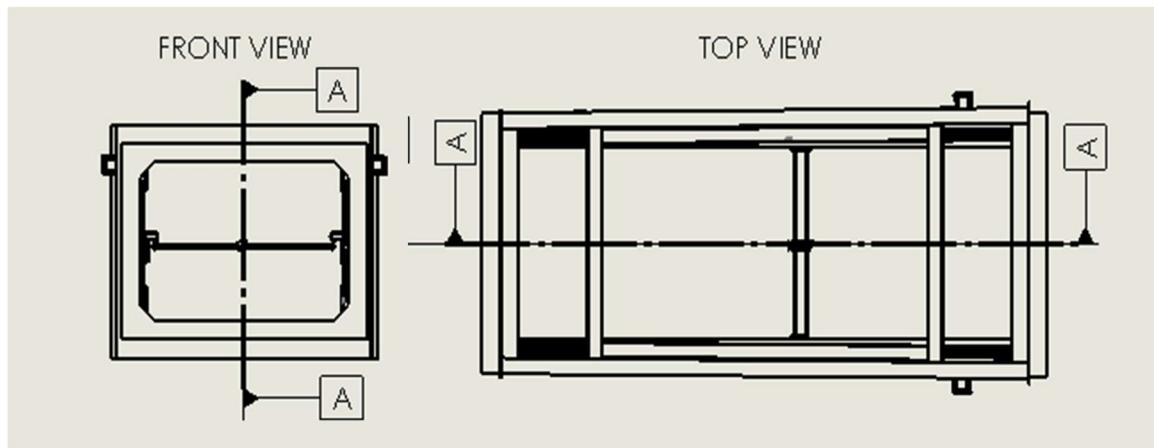


Figure 27. Datum A schematic



Figure 28. Jig used to drill dowel pin holes with hardened steel collect in center hole

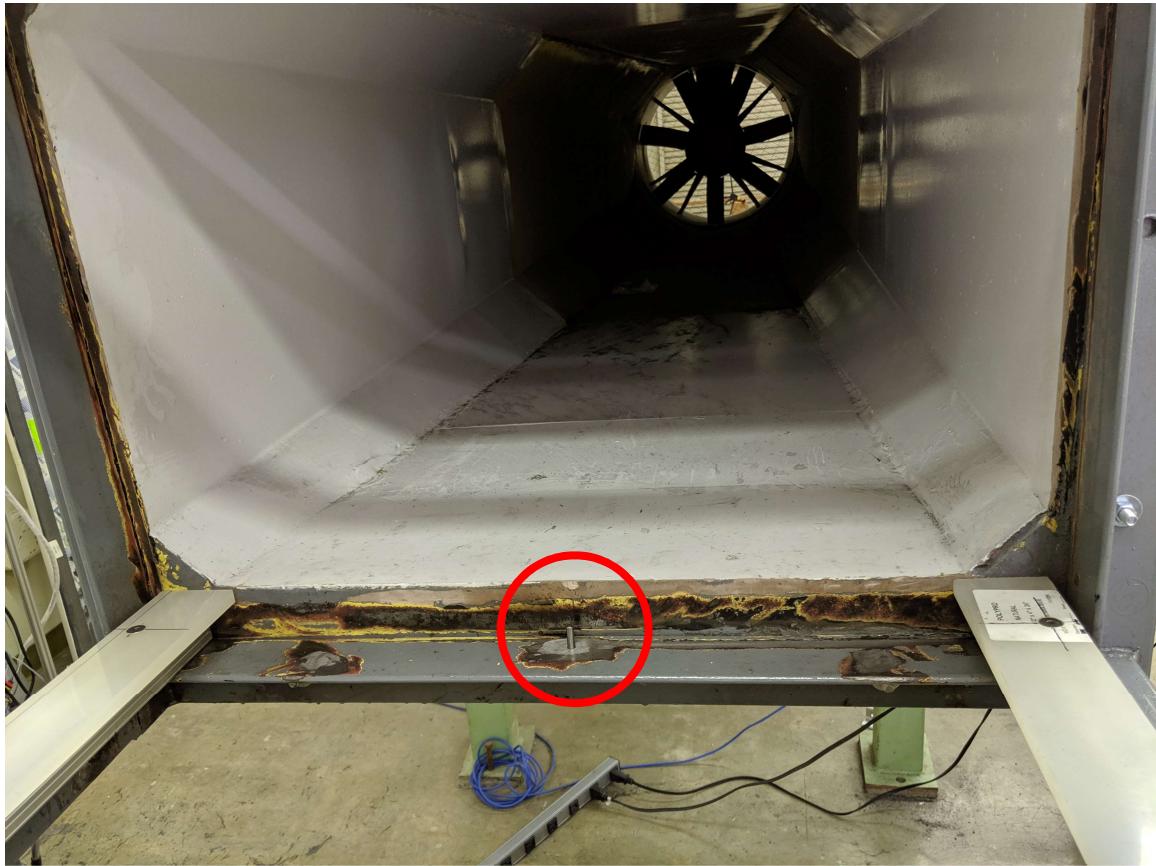


Figure 29. Installed dowel pin with chamfers installed

Traverse



Figure 30. Anechoic test section with completed traverse installed

In the wind tunnel lab, previous work has been done by Dousti and Khader [3,4] each producing a two-axis traverse for research. In the past the designs were successful, and iterative upon the previous rendition to reach its final state (Figure 30). Figure 31 shows the first iteration of the traverse by Khader [3]. This was an important first step in generating a mechanism to collect turbulence measurements via a hot wire sensor. The challenge of the test section creates is the mounting of components to a non-square metal outer frame. Additionally, this traverse generated a greater head loss due to the blockage, which was led to issues at the higher speeds of the wind tunnel.

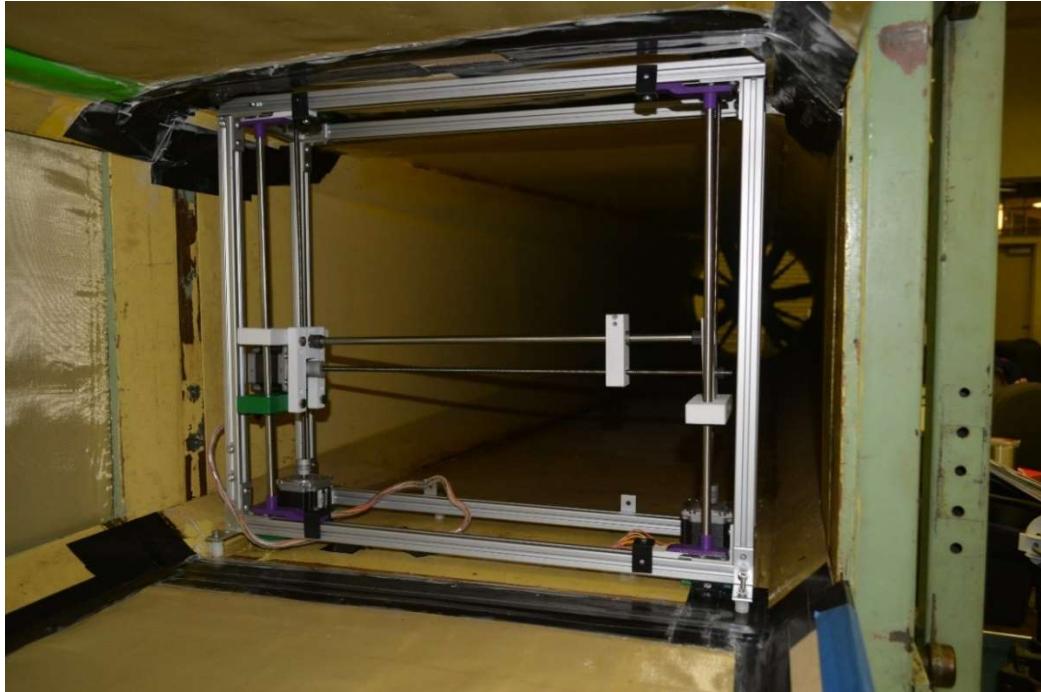


Figure 31. Two axis traverse (first revision) [3]

The second iteration by Dousti [4] was largely improved and iterated upon (Figure 32). All the motors and supporting materials were held outside the test section eliminating all but the needed components to allow the second iteration of the traverse to actuate in the horizontal and vertical axes.



Figure 32. Two axis traverse (second revision) [4]

Looking at traverse setups at other wind tunnel facilities of a similar size and setup such as the open loop Cal Poly wind tunnel brought an interesting perspective to the traverse design challenge at CSUF. The wind tunnel 3-axis traverse generated by a senior design team seen in Figure 33 and 34 [21] had all supporting and axis actuation outside the wind tunnel akin to the second to the revision. This design had numerous design parameters involved such as integration into the current frame (Figure 18), minimal flow disturbance and minimal deflection of the measurement probes, while having a cost under two thousand dollars (USD) [21].

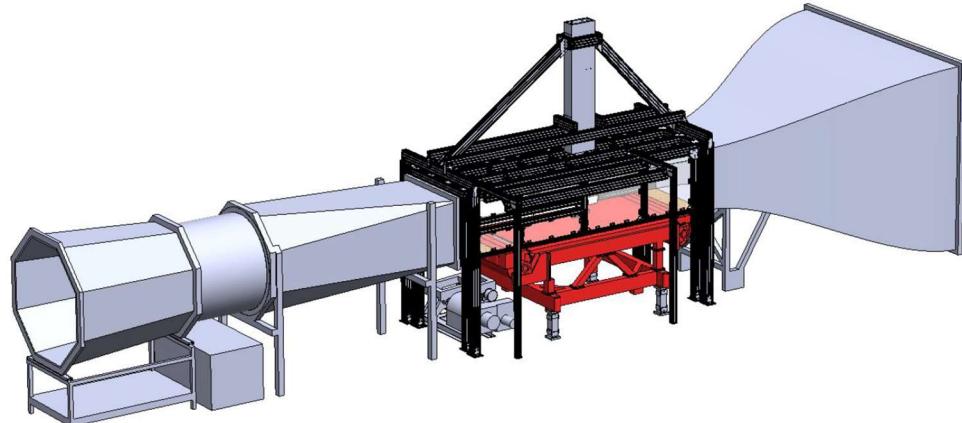


Figure 33. Cal Poly low speed wind tunnel [21]

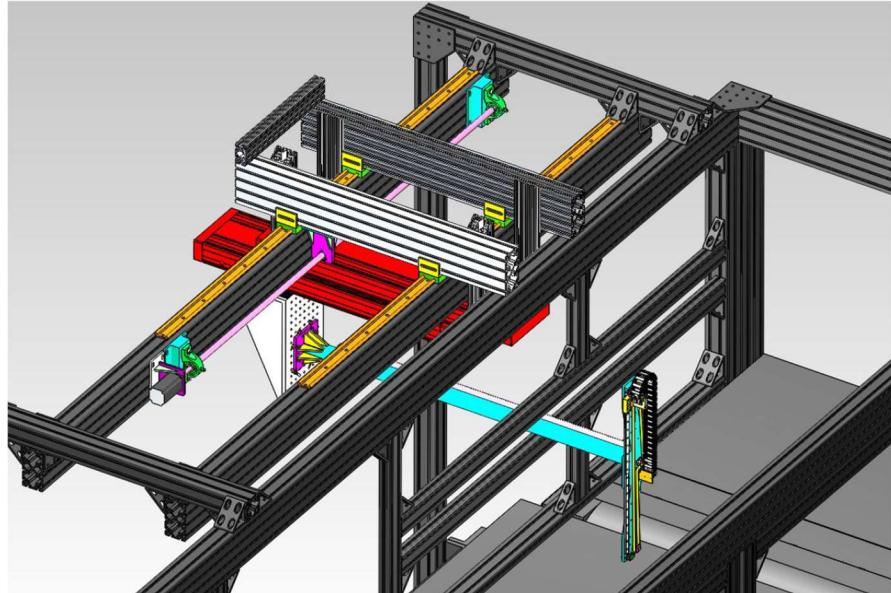


Figure 34. Cal Poly low speed 3-axis probe traverse [21]

The three-dimensional traverse (Figure 30) proved to be a difficult but rewarding design challenge. The overall concept is simple in its goal. Improve upon the two previous iterations and generate a traverse that can operate in three dimensions. Even this seemingly simple design request can cause a myriad of issues if not thought through before designing. All these issues naturally led to the workflow seen in Figure 35 by

Budynas et al. [24]. The need of a three-axis traverse was identified, and a physical design requirement (fitting inside the test section) accompanied the request. After the synthesis and analysis of numerous different solutions, a few key design factors were chosen. The largest being the railing systems would use a 20 mm extruded aluminum v-slot profile. This allowed for all gantry systems to use off the shelf, standard 3D printing parts such as GT2 timing belt, pulleys, gears, and motor shaft couplers.

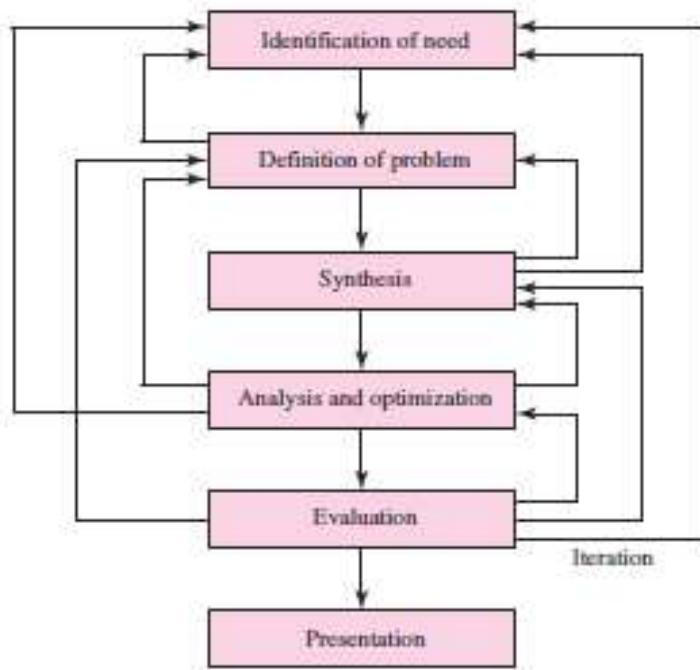


Figure 35. Iterative design process used [24]

After the railing system was decided, the X, Y, and Z axis were designed and optimized. The first challenge was how and where to mount the motors. As shown in Figure 36 and 37, the Y-axis motors are mounted outside the test section with just the shafts protruding into the test section driving the timing belts to push the main gantry (XZ axes). The Z and X motors are inside the test section and move with the gantry system. The main reasoning behind this design decision was if the Z and X gantry system

motors were stationary and mounted outside the test section, like the Y-axis motors, this would significantly increase the complexity since any belt would have to make numerous

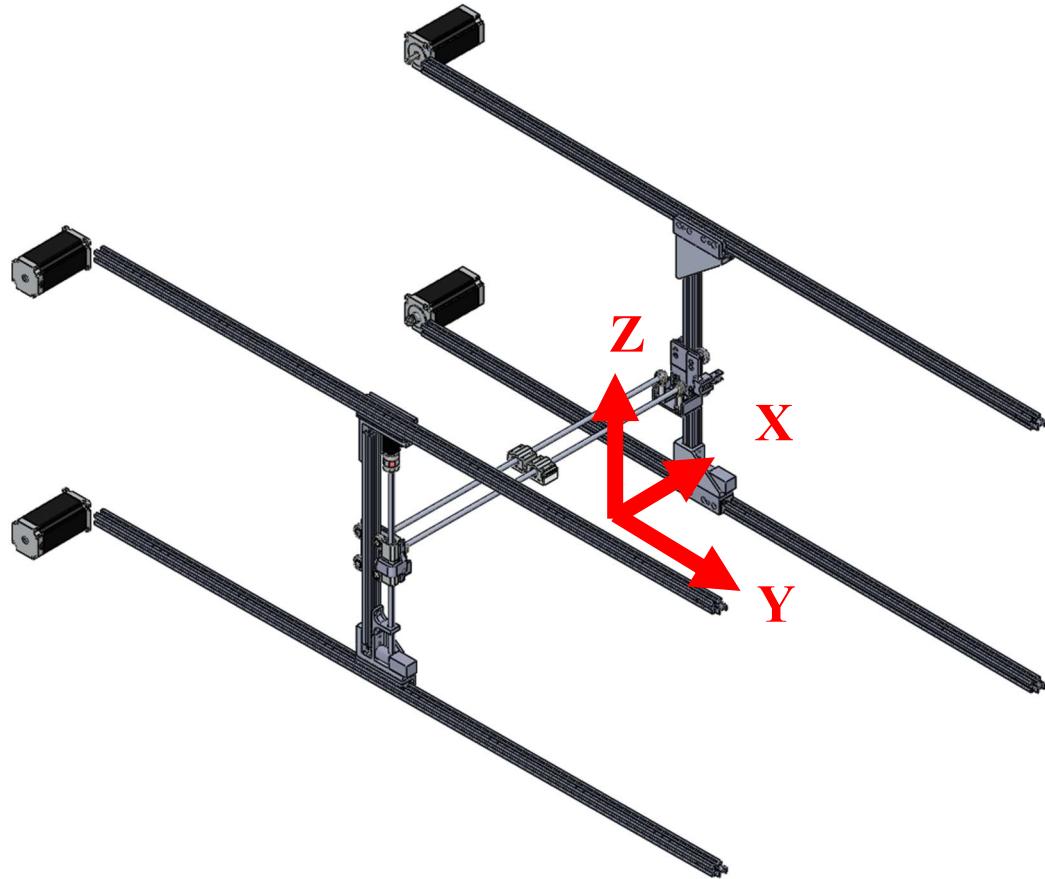


Figure 36. Traverse CAD model with labeled axes

Taking a closer look at the XZ gantry in Figure 37, the issue of the 1.5" width difference from the front to rear of the test section required some attention. The X axis rods needed the freedom to change 3/4" on each side of the test section while staying parallel with respect to each other. An eyelet turned out to be the perfect solution as its rotational freedom to allow alpha in Figure 37 to greater than or less than 90 degrees,

while allowing the X axis rods to slide freely to accommodate change in width. The X axis belt (not shown in figure 38) also required the timing belt to be kept tight with a change in length. Another simple design solution was to add a spring-loaded pulley as seen on the lower right side of Figure 37. This allows the gantry to slide along the rails while keeping the belt at a proper tension.

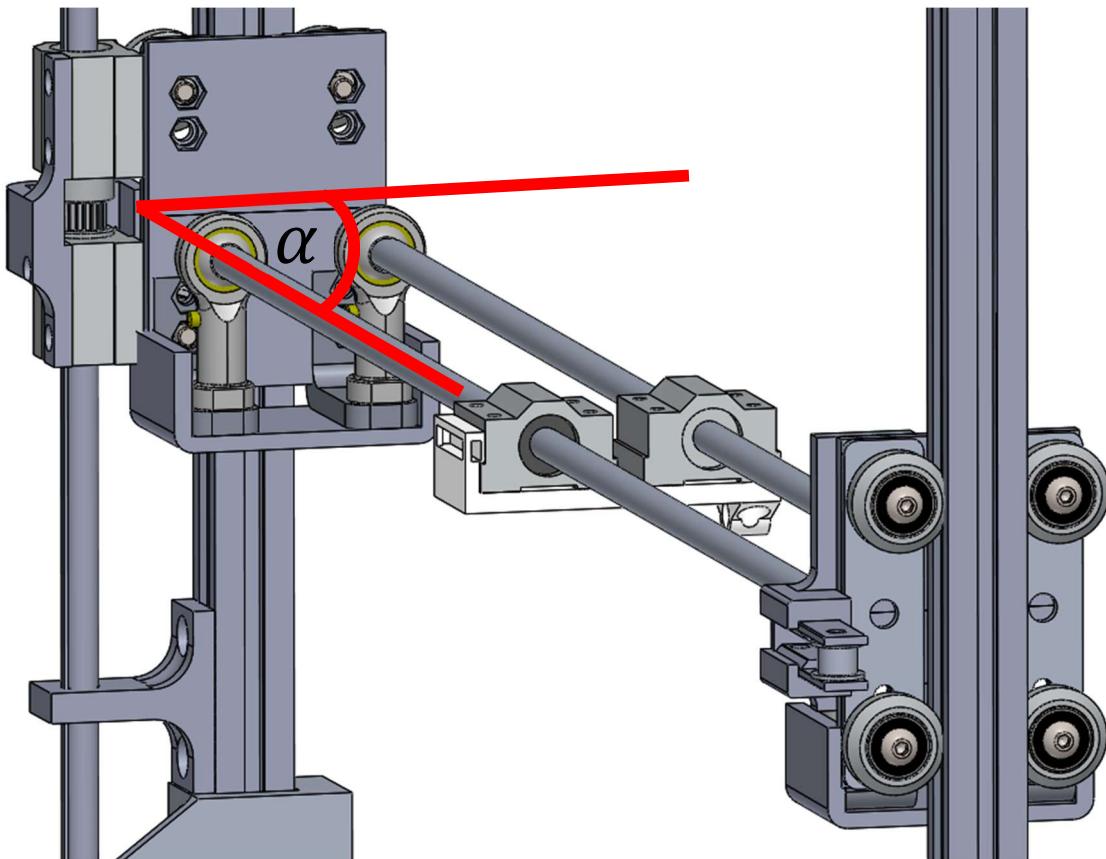


Figure 37. ZX Gantry system

The X axis gantry system (Figure 38 circled in red) is largely integrated with the XZ gantry system but have a few key stationary components. The NEMA 8 stepper motor (top center of Figure 38) turns a keyed shaft to move the X carriage along the rails via a timing belt. The system devised resulted in a keyed shaft, with two linear bearing and a GT2 timing pulley being attached to the XZ carriage. This allows the timing belt to

be in the same plane as the X linear rails, while a set screw is held in the key of the shaft forcing a rotation from the NEMA 8 motor.

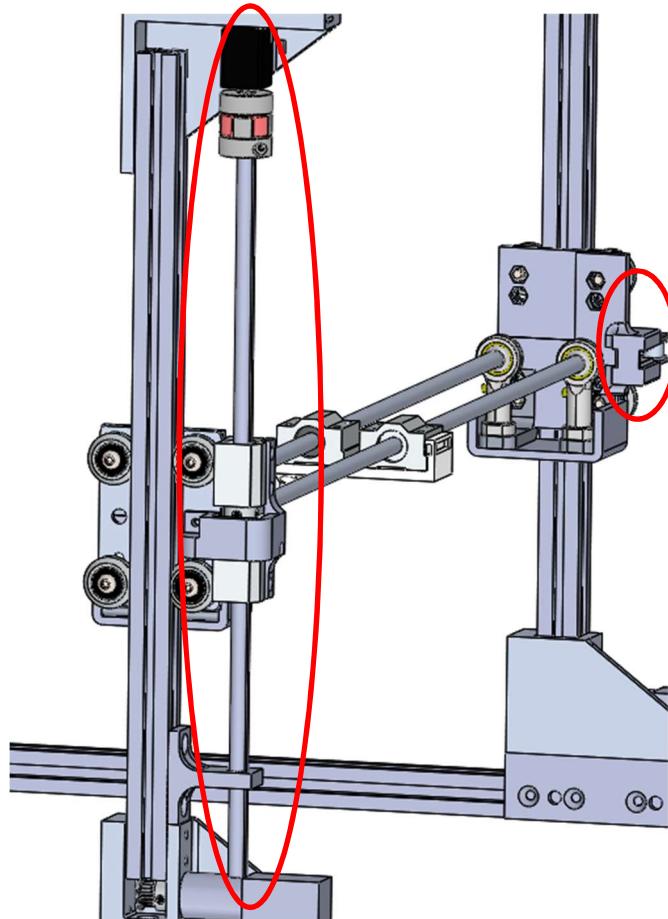


Figure 38. Static X axis assembly components.

Overall the traverse installed as seen in Figure 30 produces a very low profile while being properly installed. This section of the paper is considered largely a success in the fact a working prototype was generated and installed for future work in the wind tunnel at CSUF. Additionally, the cost of all the parts (not including any 3D printed parts) is roughly 700 USD, which could be considered a very economical option for the prototype that was produced.

CHAPTER 4

LSTM METHODS AND RESULTS

This section details all the results found of the LSTM time sequence modeling of a two microphone beamformed input, and a target of a 7 microphone beamformed signal. The overall goal of this LSTM is to accurately predict the noise suppression of a 7 microphone beamformed signal with only a signal input of 2 microphones. All the code needed along with a few sample files can be found in the Aero_ML GitHub Repository [34].

Preprocessing

The first process in creating a machine learning pipeline is to obtain any existing data for a project and import this data into a data structure for further processing and feature extraction as seen in Figure 39 [33]. For this research project, the entirety of this project was completed in MATLAB [25]. In previous data processing scripts for the transmission loss test, a batch of files were manually loaded into MATLAB, then the microphone sensitivity profiles were subtracted, along with a sound file with the wind tunnel background noise (the white noise generator turned off). This “cleaned” spectral amplitude (in dB) was then simply plotted with respect to the white noise that was produced or the transmission loss with different turbulence devices to try and increase the noise dissipation through the wake.

For this machine learning project, a traditional workflow (Figure 39) was assumed. All the .dat files from the previous acoustic wake tests were read into a three-

dimensional array, where the microphone sensitivity profiles were programmatically subtracted from a 7 microphone beamformed spectral signal and a 2 microphone beamformed spectral signal. Next, in order to prepare a predefined target, or answer, for the LSTM network, the 7-microphone signal then had the background noise file subtracted from them, while the 2 microphone beamformed signal was left as is. Both signals were then filtered using a the ‘rloess’ smoothing function in MATLAB which helps reduce any noise in the signal while preserving the shape of the data. After the data is saved into an intermediary .mat file, a random number is generated, and one of the 396 cleaned signals are plotted as seen in Figure 40 to ensure that the preprocessing step is working correctly before proceeding to the feature extraction. In Figure 40, the SPL_{denoised} signal is the unsmoothed 7-microphone beamformed signal with the background noise subtracted. This is plotted along with the smoothed 2 and 7-microphone beamformed signals.

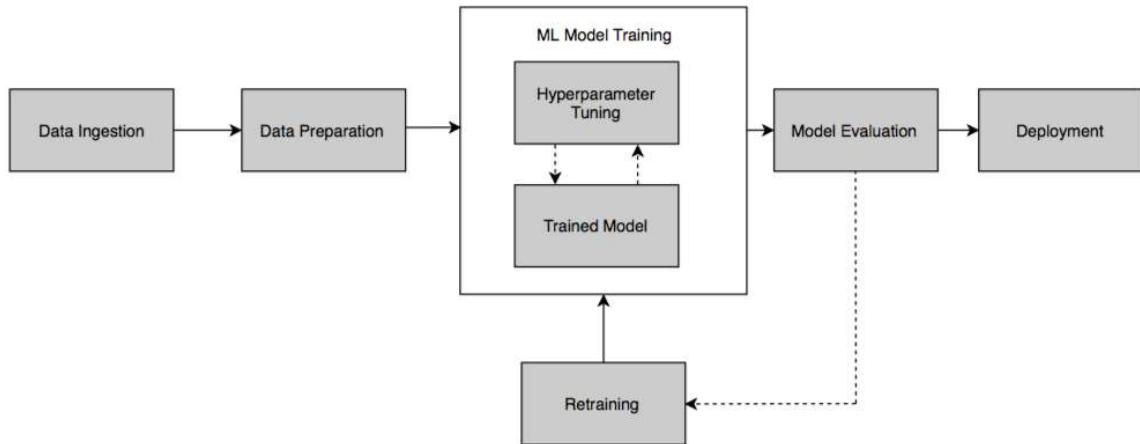


Figure 39: Simple machine learning pipeline [33]

Still following Figure 39, features must further be extracted from the signals before being fed into the LSTM network architecture. The reasoning for this is to help give the neural network parameters that are easier to identify changes in rather than

feeding in a whole signal, which would introduce noise that would confuse the network, leading to poor performance. In LSTM networks with audio, it is common practice to use the Mel-frequency cepstrum spectrogram, a traditional spectrogram, or coefficients produced from either of these algorithms. The theory behind how these items will be further explained, however it is worth noting that all three types of signal representations were used in training in comparison to one another to see if one offered better performance over another (little to no difference was detected).

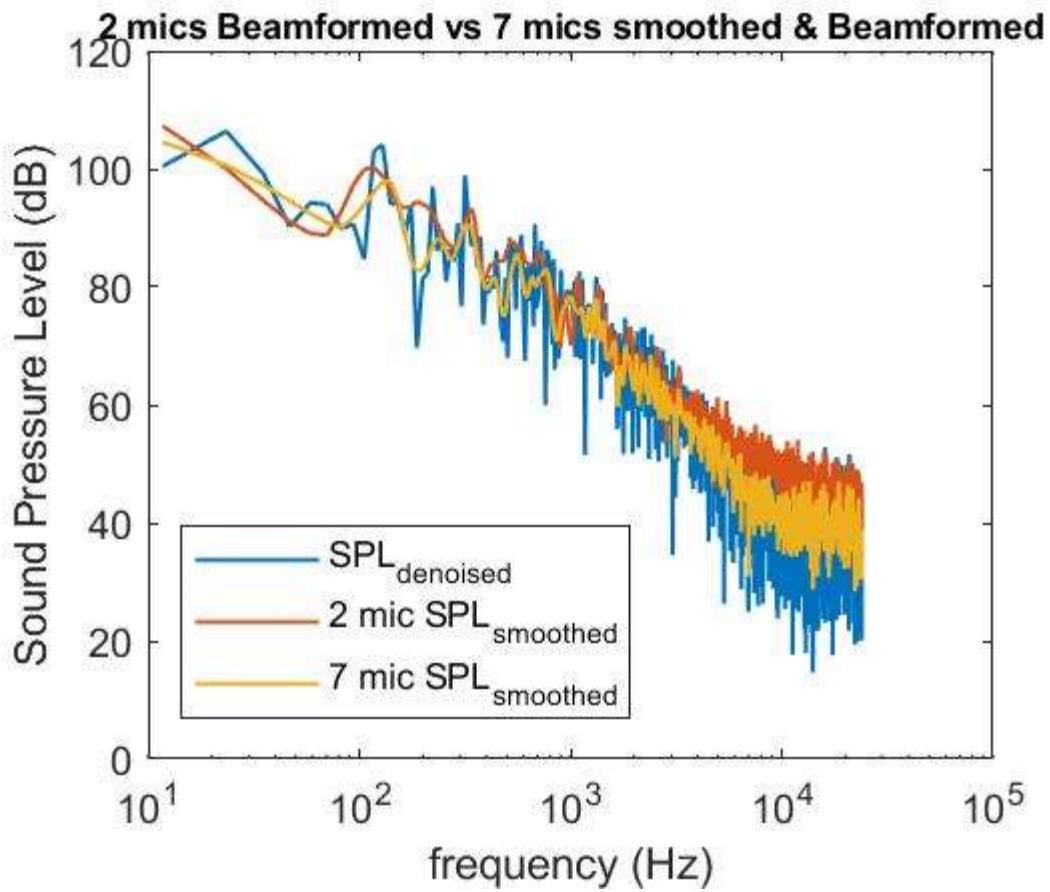


Figure 40. Example output of the LSTM preprocessing MATLAB script

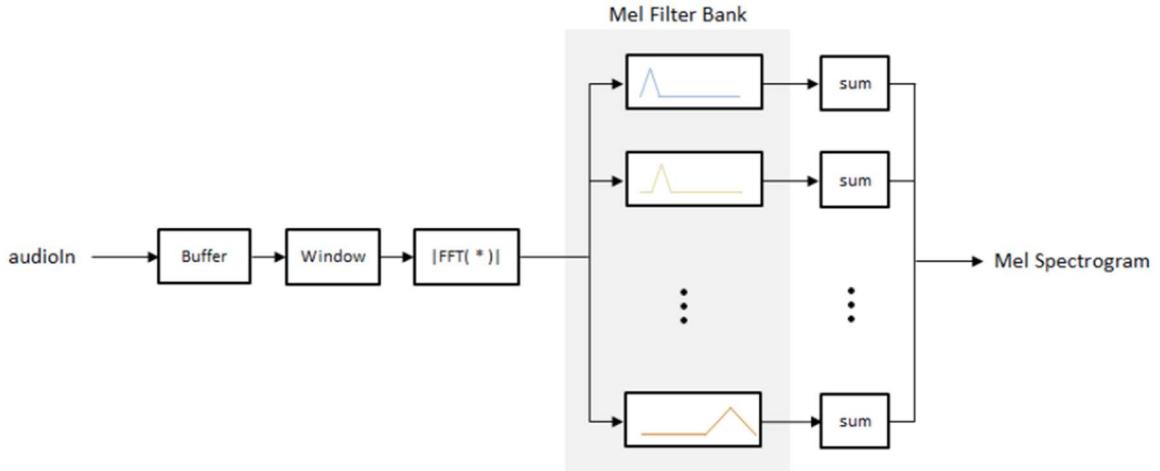


Figure 41. Mel spectrogram function MATLAB [35]

Mel spectrograms are a spectrogram that considers how humans perceive different frequency ranges. As seen in Figure 41, the Mel spectrogram function in MATLAB [35] starts with an audio input, where a window length and overlap amounts are calculated based on the length of the input signal. A periodic hamming window is then applied to each frame, where the frame is converted to the spectral domain using the Fast Fourier Transform (FFT) algorithm. The main distinguishing factor between a typical spectrogram and a Mel spectrogram is the series of band pass filters the frequency bands are passed through. As seen in Figure 42, these half-overlapped triangular filters are bands are more densely located on frequencies that humans are more sensitive to than the higher frequencies, which on the Mel scale are equally spaced. Upon visual inspection of Figure 43, there is a difference in the Mel Spectrogram produced by the 2 and 7-microphone beamformed signals, which is another good logic check before setting up these calculated coefficients into the LSTM network.

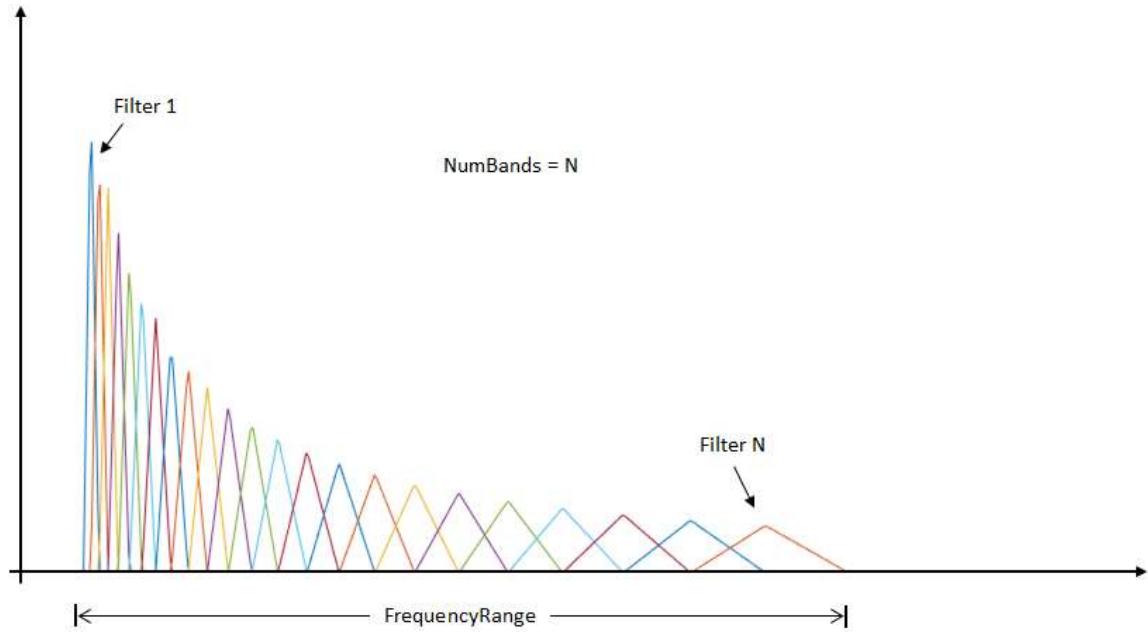


Figure 42. Mel filter bank design [35]

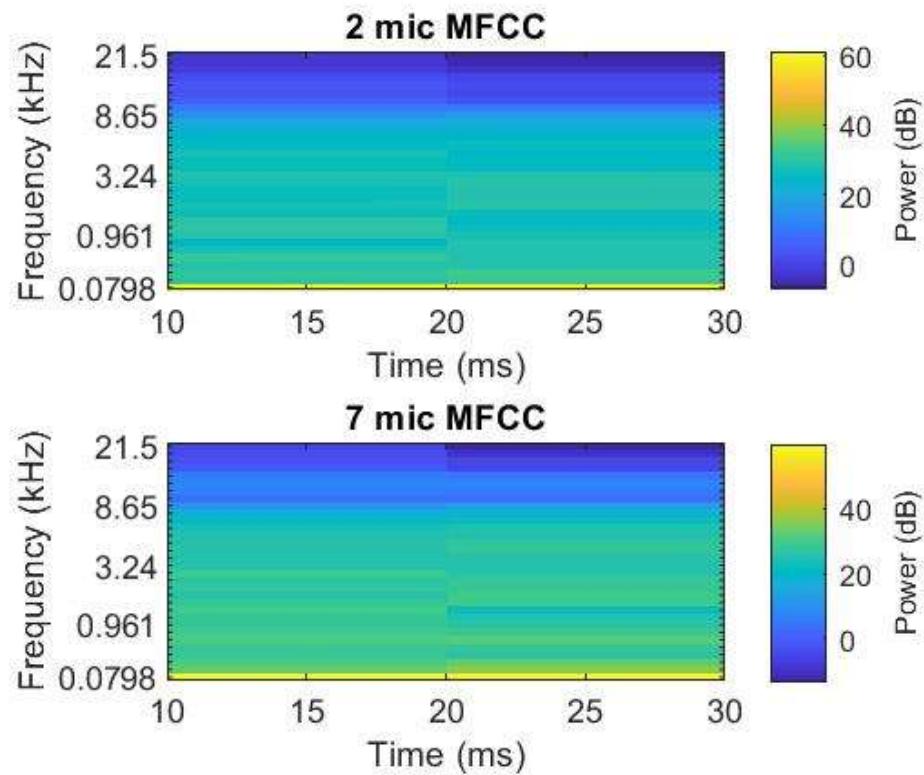


Figure 43. Mel-frequency Spectrogram

LSTM Network Architecture and Results

Following the next steps of Figure 33, now that all the input and target features had been calculated and organized into a cell array in MATLAB, it was ready to be fed into an LSTM network architecture. In attempt to find the best network architecture, over 50 different variations were attempted and trained, not including different hyperparameter tuning that was done. All machine learning models were trained on the old data produced by Dousti and Khader [3,4] and predictions are made on the new set of validation data. This is usually the best test for a model that produces a sequence of data as the output. Traditional classification networks are much easier to evaluate the performance of the network. The number of correct classes are compared to incorrect, and a concrete number can be produced. For sequence to sequence models, the model performance and evaluation is not as explicit, although there are a few standard metrics that helps indicate the “goodness” of fit of a model. These more standard metrics are plotted and discussed below.

The results visualized in Figures 44-47. All results were trained on a Nvidia GeForce GTX 1080 with 8 Gb of dedicated GPU memory, 2560 CUDA cores at 1771.5 MHz clock rate. As previously mentioned, evaluating a regression output type machine learning model can be more ambiguous than a classification model. This means that since the LSTM model is outputting a 2048 sequence of data, thinking of a prediction sequence as “correct” or “incorrect” is not the proper terminology to use. One way to evaluate an LSTM model would be the RMSE value that is calculated at the end of each epoch (or one complete presentation of the data set to be learned by the model). The RMSE as seen

in equation 2 can often be thought of as the variance between what the model is outputting and the target values being predicted.

$$RMSE = \sqrt{\frac{1}{n} \sum_{k=0}^n (Predicted_i - Actual_i)^2} \quad (2)$$

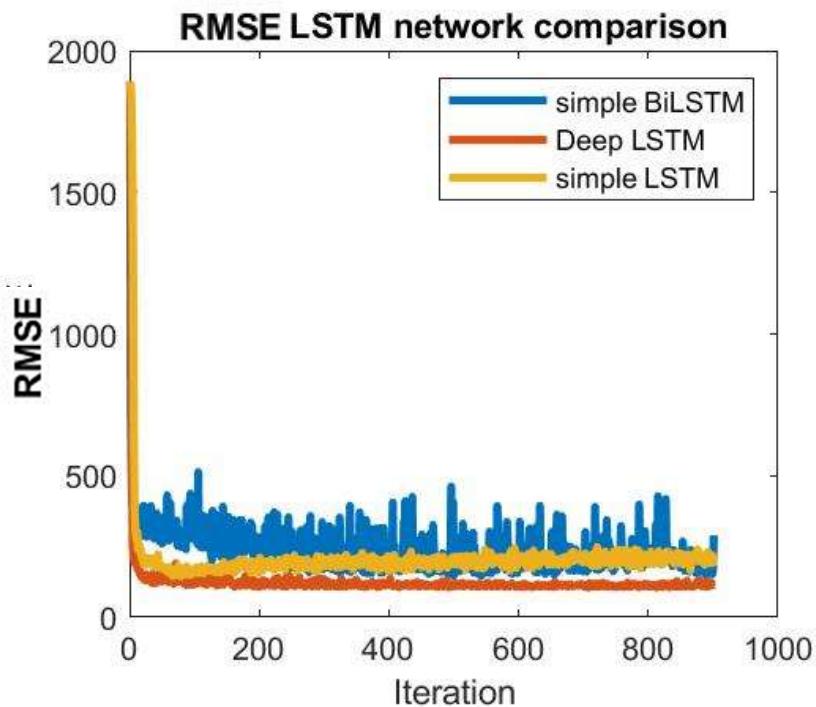


Figure 44. RMSE LSTM network comparison graph

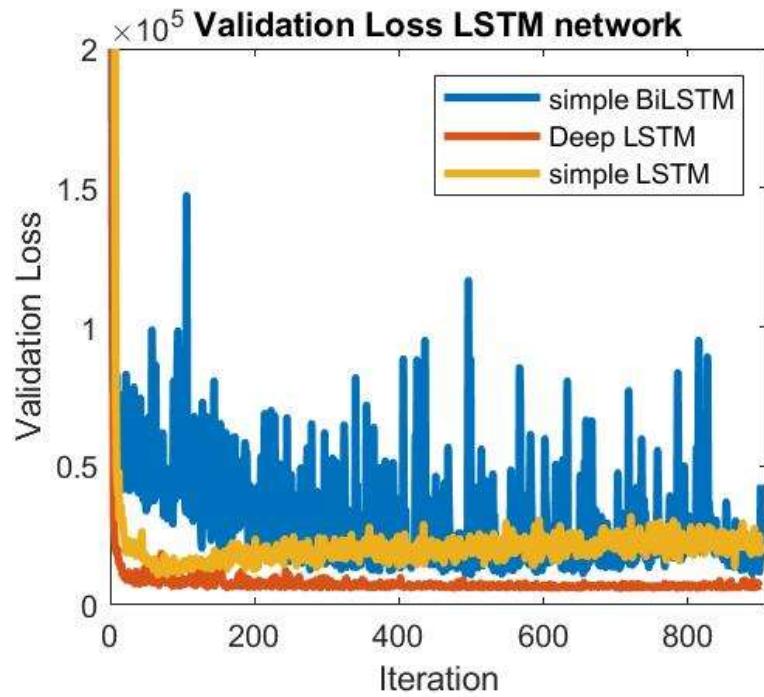


Figure 45. Validation loss LSTM network comparison graph

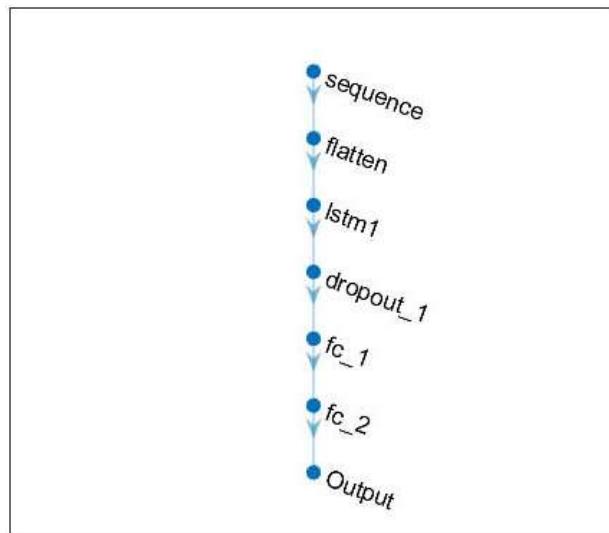


Figure 46. A simple LSTM network architecture

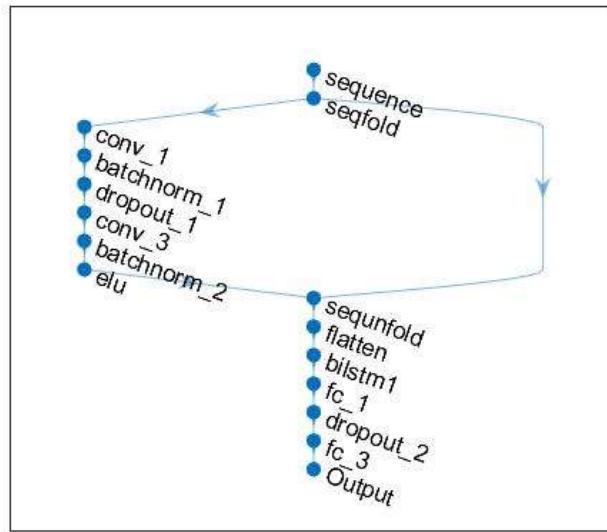


Figure 47. A Deep C-BiLSTM network architecture

As seen in Figure 44, the three main model architectures that were investigated was a simple LSTM model, simple BiLSTM, and Deep C-BiLSTM. The simple LSTM (Figure 46) has a few fully connected layers to allow the model to learn everything it can from the relatively small data set it was given, and dropout layers which randomly delete neurons attempting to have the model not overfit the data set. A BiLSTM model is simply an LSTM logic gate that convolves over the sequence forward and in reverse, in another attempt to not have the BiLSTM overfit to a data set. If a model is overfit to a data set, when new predictions with a completely dependent data set are made, the accuracy would sharply drop, defeating the purpose of the machine learning model which is supposed to learn general nonlinear features of the data set, so when new data is fed in, the predictions still remain fairly accurate. Lastly a deep C-BiLSTM (Figure 47) is simply a BiLSTM network with a traditional CNN attached to the network. A deeper network would in theory allow the model to learn more about a given data set, although as seen in Figure

44, the Deep C-BiLSTM did in fact outperform the other networks, but by a marginal amount.

Looking at the validation loss graph (Figure 45) this metric is the forward pass of the network in which it sums all the calculated errors on the training and validation set of data. While training the validation set of data is a percentage of the training data that the model uses to check how well the model is training, not to be confused with the validation set of data that was produced independently for this research and helps best prove the generalizability of a machine learning model. Again, the Deep C-BiLSTM outperformed the other models but took roughly three times as long to train.

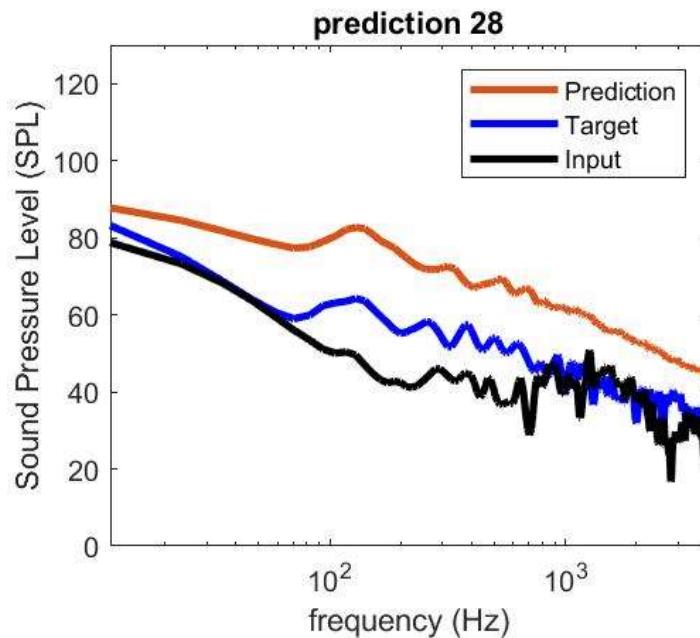


Figure 48. Prediction 28 of a simple LSTM

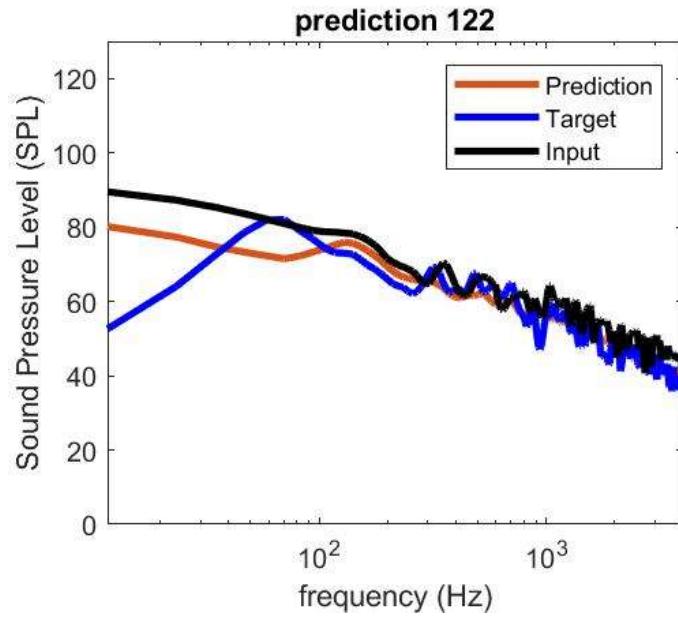


Figure 49. Prediction 122 of a simple LSTM

As seen in Figures 42 and 43, the LSTM can properly generate a prediction (Prediction line, orange) of what a 7 microphone beamformed signal would look like (Target line, black) based off of a 2 microphone beamformed input (Input) . As the model will sometimes do a better job (Figure 49), while are off by a considerable amount (Figure 48) in other predictions. As these are predictions made by the model on data that was never seen before, it offers valuable insight as to how much the model understands, or generalizes, of the data it has been given. Considering that the SPL is in decibels, the model will do a decent job at finding the best fit line statistically, but will omit details that are critical, such as the larger overall magnitude of the new prediction as seen in figure 42.

This is a first step in an application to machine learning in the wind tunnel lab, and with further exploration could offer better results in the future with more data and different machine learning approaches. A theorized different approach to yield a better fit

prediction and generalization of the data sets would be to break up the very long 2,048 element sequence into two or more parts and trained by different models. This could, in theory, allow each model to learn more details about the lower and higher frequency bands. Although few other tactics to improve prediction accuracy could be theorized, this would be outside the scope of this project, and would be left to future research.

CHAPTER 6

CONCLUSION

Wind Tunnel Renovations

In conclusion the modularity of the wind tunnel was regarded a success in the right step in assisting future research. The datum was installed with the chamfers correctly placed in the wind tunnel. The new anechoic chambers and Kevlar panels were installed for the validation anechoic data, while ensuring the wind tunnel setup could be changed a little easier. Standard panel dimensions were generated and used for simpler top and bottom panel frame geometry while doubling the dowel pin as a locating feature on the frames.

Traverse

The traverse was a large step for the wind tunnel lab for further enabling quantitative measurements in the flow field of further tests in the wind tunnel. The 3-dimensional traverse prototype was installed and needed a small amount of programming to have it fully up and running. Overall the design solution was found to a problem that required everything to have strict limits to fit inside the test section, and multiple degrees of freedom specifically in the XZ gantry system.

LSTM

The machine learning portion of this research paper was some of the most exciting research, as this specific task that was completed is largely uncharted for the fields of signal processing and time-series based prediction models. This topic could lead to very useful software in the Wind tunnel lab and is a complete working prototype. Resulting full sized predictions were not the most ideal in some cases, but there are many tactics that could be attempted in the future to increase the understanding and accuracy of prediction models. One that quickly comes to mind is having multiple models each predicting part of the entire SPL signal, as 2,048 point is extremely long for one model to predict. Another attribute to the project that could be used in the future is the GitHub [24] code repository that could be easily used and modified for further research.

REFERENCES

1. Baals, Donald D., and William R. Corliss. Wind Tunnels of NASA. NASA, 1981.
2. Dousti, Niayesh."Effects of Trailing Edge Serrations on Shielding of Aeroacoustic Noise Sources by a Momentum Wake" 2019. Print.
3. Khader, Syed Zeeshan,."Development of an Anechoic Wind Tunnel". 2017. Print.
4. S. Mayoral, "Aerodynamic Forces on Immersed Bodies", California State University, Fullerton, 2016.
5. Swift, H., "A Review of the Literature Related to Potential Health Effects of Aircraft Noise," PARTNER Project 19 Final Report, Purdue University, July 2010
6. Casalino, D., Diozzi, F., Sannino, R., and Paonessa, A., "Aircraft noise reduction technologies: A bibliographic review," *Aerospace Science and Technology*, Vol. 12, 2008, pp. 1–17.
7. "3D CAD Design Software.", Product Categories, www.solidworks.com/.
8. Guo, Y., Burley, C. L., and Thomas, R. H., "On Noise Assessment for Blended Wing Body Aircraft," 52nd Aerospace Sciences Meeting, AIAA SciTech Forum,AIAA Paper 2014-0365, 2014.
9. Overview of the NASA Environmentally Responsible Aviation
ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20140012638.pdf.
10. Papamoschou, D., and Mayoral, S., "Jet Noise Shielding for Advanced Hybrid Wing-Body Configurations," 49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, AIAA Paper 2011-912, 2011.
11. "Boeing Frontiers Online." Boeing,
www.boeing.com/news/frontiers/archive/2005/december/ts_sf07.html.
12. Devenport, W. J., Burdisso, R. A., Borgoltz, A., Ravetta, P. A., Barone, M. F., Brown, K. A., and Morton, M. A., "The Kevlar-walled anechoic wind tunnel," Journal of Sound and Vibration, Vol. 332, No. 17, 2013, pp. 3971–3991.

13. Remillieux, M. C., Crede, E. D., Camargo, H. E., Burdisso, R. A., Devenport, W. J., Rasnick, M., Seeters, P. V., and Chou, A., "Calibration and Demonstration of the New Virginia Tech Anechoic Wind Tunnel," 14th AIAA/CEAS Aeroacoustics Conference, AIAA Paper 2008-2911, 2008.
14. Voices, Valley. "Cutting Through The Machine Learning Hype." Forbes, Forbes Magazine, 16 Nov. 2016, www.forbes.com/sites/valleyvoices/2016/11/16/cutting-through-the-machine-learning-hype/#5092975a5465.
15. Convolutional Neural Network., Convolutional Neural Network - MATLAB & Simulink, www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html.
16. Hegde, Vishakh, and Sheema Usmani. "Parallel and Distributed Deep Learning - Stanford.edu." Parallel and Distributed Deep Learning, stanford.edu/~rezab/classes/cme323/S16/projects_reports/hedge_usmani.pdf
17. Hochreiter, Sepp. "The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions." International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 06, no. 02, 1998, pp. 107–116., doi:10.1142/s0218488598000094.
18. Rojas Raúl. Neural Networks: a Systematic Introduction. Springer, 2000. http://www.inf.fu-berlin.de/inst/ag-ki/rojas_home/documents/1996/NeuralNetworks/neuron.pdf
19. Deloche, François. "File:Recurrent Neural Network Unfold.svg." File:Recurrent Neural Network Unfold.svg - Wikimedia Commons, commons.wikimedia.org/wiki/File:Recurrent_neural_network_unfold.svg.
20. "Long Short-Term Memory (LSTM) Networks." Networks - MATLAB & Simulink", www.mathworks.com/discovery/lstm.html.
21. Boonya-anata, Mel, et al. "3-Axis Automated Probe Traverse for Aerodynamic Testing." Digital Commons, digitalcommons.calpoly.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=1378&context=mesp.
22. Labview. National Instruments, 1997.
23. MATLAB 2019a, The MathWorks, Inc., Natick, Massachusetts, United States.
24. Budynas, R. G., & Nisbett, J. K. (2015). Shigleys mechanical engineering design., pp. 4,6 New York, NY: McGraw-Hill Education.
25. Machine Learning in MATLAB., Machine Learning in MATLAB - MATLAB & Simulink, www.mathworks.com/help/stats/machine-learning-in-matlab.html.

26. Raschka, S. (2015, March 24). Single-Layer Neural Networks and Gradient Descent. Retrieved from https://sebastianraschka.com/Articles/2015_singlelayer_neurons.html
27. Veen, Fjodor van. “The Neural Network Zoo.” The Asimov Institute, 27 Apr. 2019, www.asimovinstitute.org/neural-network-zoo/.
28. Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the “brain.Psychological Review”, 65(6), 386-408. doi:10.1037/h0042519
29. Karpathy, A. (2015, May 21). The Unreasonable Effectiveness of Recurrent Neural Networks. Retrieved July 28, 2019, from <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>
30. Awad, M., & Khanna, R. (2015). “Efficient learning machines: Theories, concepts, and applications for engineers and system designers.” Berkley?: Apress Open.
31. 10 Companies Using Machine Learning in Cool Ways., WordStream, www.wordstream.com/blog/ws/2017/07/28/machine-learning-applications.
32. Raschka, Sebastian. “Activation Functions for Artificial Neural Networks.” Activation Functions for Artificial Neural Networks” - MLxtend, rasbt.github.io/mlxtend/user_guide/general_concepts/activation-functions/.
33. AutoML for building simple to complex ML pipelines. (2018, September 21). Retrieved from <https://hub.packtpub.com/automl-build-machine-learning-pipeline-tutorial/>
34. https://github.com/andrewbartels1/Aero_ML
35. MelSpectrogram - mathworks. (n.d.). Retrieved from https://www.mathworks.com/help/audio/ref/melspectrogram.html#mw_3b18aac-c275-466a-ae6a-a16e80ff766d
36. Mayoral, Salvador, and Ramitha Edirisinghe. “On the Mechanical Design of the Rolling Road Addition to the CSUF Wind Tunnel.” On the Mechanical Design of the Rolling Road Addition to the CSUF Wind Tunnel, 2 Apr. 2019, www.sae.org/publications/technical-papers/content/2019-01-0651/.

APPENDIX A: SUPPLEMENTAL FIGURES



FIGURE A1: Low Speed Wind Tunnel Fan Nameplate

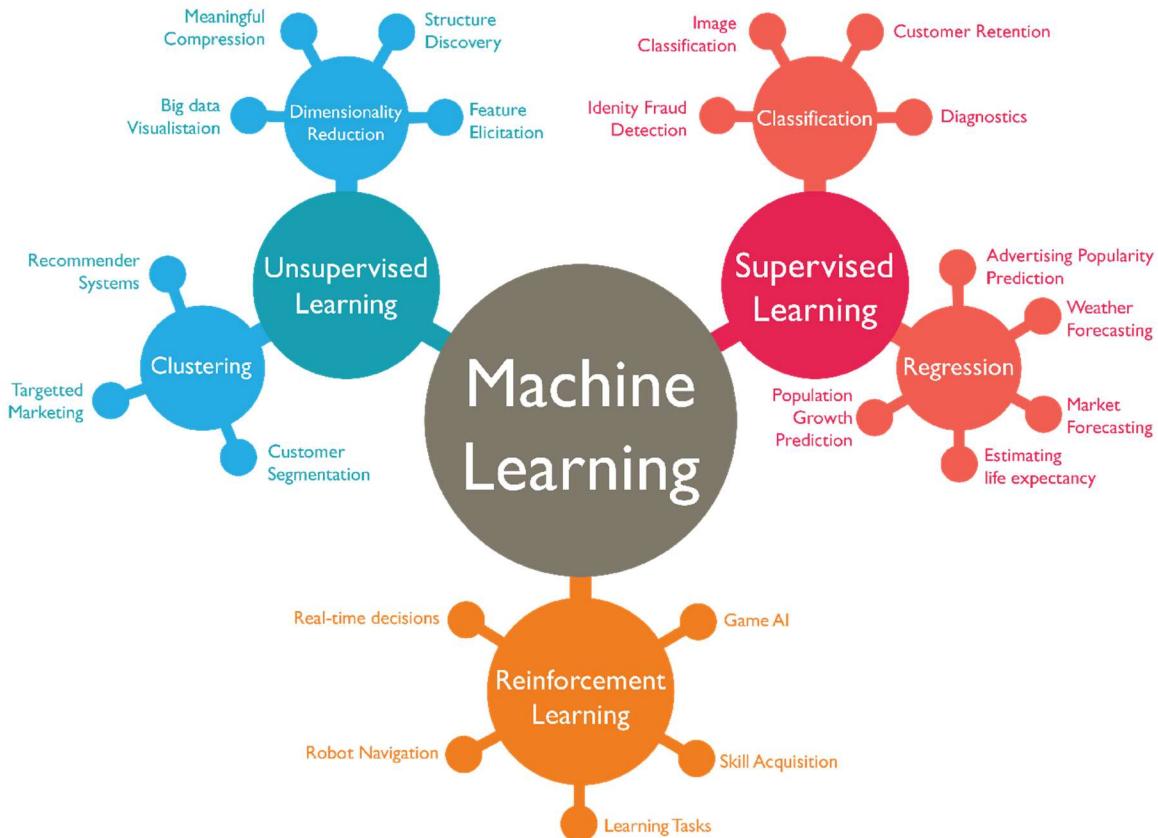


Figure A2. Machine Learning Applications [24]

Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	Multi-layer NN	

Figure A3. A few activation functions[25]

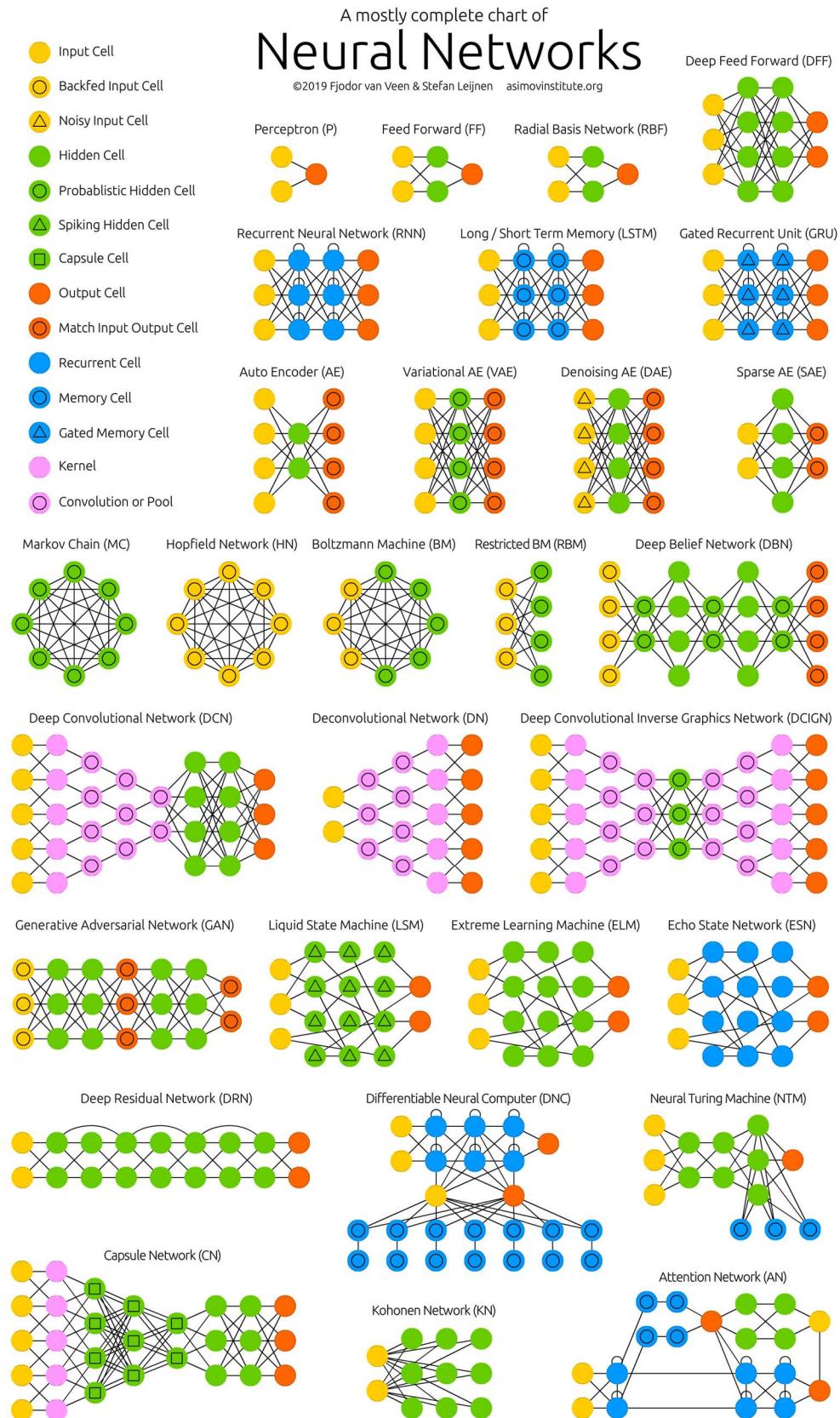


Figure A4. A mostly complete neural network architecture chart [26]

