Development of a Vertical Axis Wind Turbine (VAWT) Field Research Center to Investigate the Inner-Rotor Fluid-Structure Interactions

A THESIS

SUBMITTED TO THE FACULTY OF THE

UNIVERSITY OF MINNESOTA

BY

JUSTIN TODD RUEDEBUSCH

IN PARTIAL FULFILLMENT OF THE REQUIERMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

ADVISOR:

CRAIG S. HILL, Ph.D.

COMMITTEE MEMBERS:

ALISON B. HOXIE, Ph.D.

DAVID A. ANDERSON, Ph.D.

MAY 2023

Acknowledgements

I would like to thank my advisor, Dr. Craig Hill, of the Mechanical and Industrial Engineering department at the University of Minnesota Duluth (UMD). He has always been available for questions regarding research, homework, and guidance on my academic and professional career.

I would also like to thank Dr. Alison Hoxie for her work done on the original wind turbine at the UMD Research and Field Studies Center and her guidance and help throughout my academic career. Both Dr. Hill and Dr. Hoxie for inspiring, encouraging and supporting me pursue my interests in renewable energy! I'd also like to thank each of Dr. Hill, Dr. Hoxie and Dr. David Anderson for the support and commitments made to be a part of my thesis committee.

Finally, I would like to thank my wife, family, and close friends for the unconditional support throughout this journey, for engaging in conversation with my interest and providing encouragement during the difficult times faced in the last several years!

Dedication

This dissertation is dedicated to my late, current, and future family for a happy and healthy life.

Abstract

A state-of-the-art vertical axis wind turbine (VAWT) research facility has been developed at the University of Minnesota Duluth Research and Field Studies Center. The focus of this project is to develop a facility that is designed to address the deficit of field scale VAWT studies and provide a physical investigation to the characteristics of VAWTs, including the inner-rotor fluid dynamics and rotor-scale fluid-structure interactions. The research is driven by the outlook for renewable energy and the adoption of VAWTs for distributed wind, array clustering and offshore energy, among other applications.

A 2.4kW H-Darrieus VAWT is mounted on a 19.8m (65ft) steel tower that is fitted with 3D sonic anemometers and meteorological sensors. There are three measurement stations at 6.1m (20ft), 12.2m (40ft) and 18.3m (60ft), which sample up to 32Hz. A high-resolution 3D sonic anemometer positioned inside the rotor location at 18.3m (60ft) samples up to 100Hz to capture 3D wind components enabling the potential for detailed vortex dynamic characterization. Atmospheric conditions are also sampled at each station at 1Hz. An initial investigation into the inner-rotor fluid dynamics illustrates the presence and ability to measure vortex dynamics and shedding from turbine blades. While this specific project designed the facility, the complex multi-parameter data acquisition system, and provided initial insights into VAWT inner-rotor vortex dynamics, a deeper investigation of this phenomenon should be conducted to evaluate how performance characteristics of the turbine are affected. Additionally, consideration of other atmospheric conditions should be considered alongside the inner-rotor fluid-structure interactions to expand the understanding of VAWT performance regions that experience large differences in meteorological conditions from season to season.

The site experiences a vast number of atmospheric conditions. These conditions include extreme winds, icing events including rain, sleet and snow with historical maximum and minimum temperatures of 41.1°C (106°F) and -40.6°C (-41°F), respectively. It is also easily accessible providing the ability to be a cross-functional facility with education, outreach, and cross-disciplinary research with a focus to provide real-time open access data. Given the extreme conditions and ease of accessibility, the site is well suited to provide contributions to VAWT research and design.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

VAWT	Vertical Axis Wind Turbine
HAWT	Horizontal Axis Wind Turbine
UMD	University of Minnesota Duluth
DAQ	Data Acquisition
AGL	Above Ground Level
WEO	World Energy Outlook
IEA	International Energy Agency
AEO	Annual Energy Outlook
EIA	Energy Information Administration
STEPS	Stated Policies Scenario
SDS	Sustainable Development Scenario
APS	Announced Pledges Scenario
NZE	Net Zero Emissions by 2050
AoA	Angle of Attack
TSR	Tip Speed Ratio
CFD	Computational Fluid Dynamics
PIV	Particle Image Velocimetry

1. Introduction

1.1. Field Scale VAWT for Fluid Structure Interactions

A field-scale experimental facility used to evaluate fluid structure interactions of a vertical axis wind turbine (VAWT) has been developed at the University of Minnesota – Duluth (UMD). The turbine and corresponding experimental equipment are located at the UMD Land Labs (i.e., UMD Research and Field Studies Center) just North of Duluth, MN (46.871758, -92.057184) (Figure 1.1).

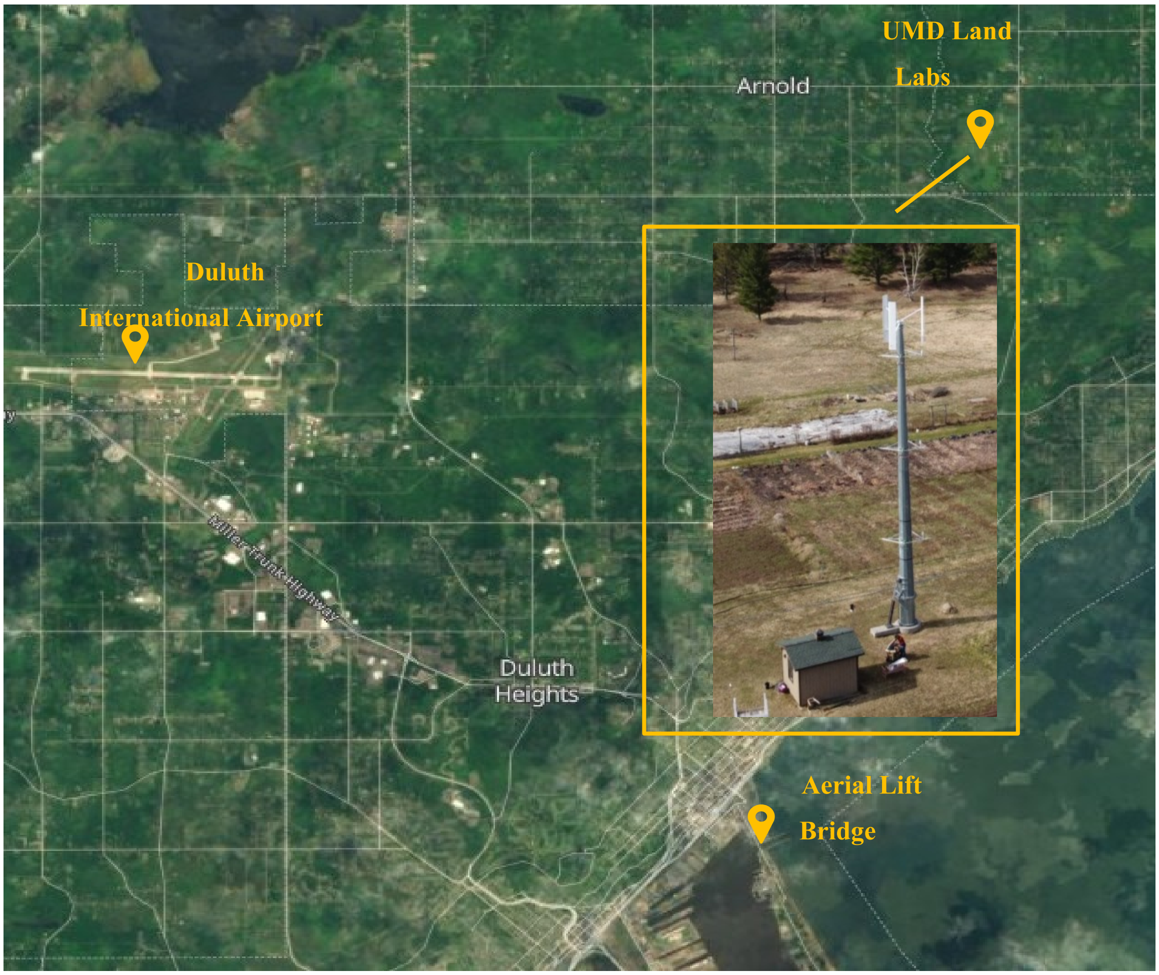


Figure 1.1: UMD Land Labs VAWT with the Duluth International Airport and Aerial Lift Bridge for reference as seen from Google Earth.

The tower is 19.8 meters (65 feet) in height, outfitted with various meteorological sensors to classify the wind resource at the site and energy potential of the turbine by examining the fluid-structure interactions. This setup is distinguished by its Semtive Nemoi M 2.4kW VAWT with a Campbell Scientific CSAT3BH heated three-dimensional anemometer located inside the rotor approximately 0.51 m (20 in) radially inward from the lower blade tip. This unique setup is intended to be used as a cross-functional facility for education, outreach, and research. The tower is relatively easy to lower providing the ability to adjust for research or give educational demonstrations (Figure 1.2). Data from the facility can be applied to a variety of fields (e.g., hydrology, agriculture, among many others) and is easily accessible for UMD students, staff, and faculty, while being in close proximity to public spaces frequented by local residents.

Figure 1.2: Tower lowered for installation of the turbine and sensors.

Wind energy is a rapidly growing field with many companies and countries targeting net-carbon-zero energy by 2050. Horizontal axis wind turbines currently dominate this sector, but a renewed focus has been put on VAWTs due to their unique characteristics. Möllerström et al. (2019) has illustrated the revival of VAWTs, specifically of the H-Darrieus type in Europe (Figure 1.3). The revival of these turbines has been attributed to a variety of factors including, but not limited to, offshore wind energy (McCabe et al., 2022; Möllerström et al., 2019; Zhao et al., 2022), array clustering to enhance performance (Brownstein et al., 2019; Strom et al., 2022), distributed energy applications (McCabe et al., 2022) and advancements in materials and manufacturing (McCabe et al., 2022; Möllerström et al., 2019; Zhao et al., 2022).

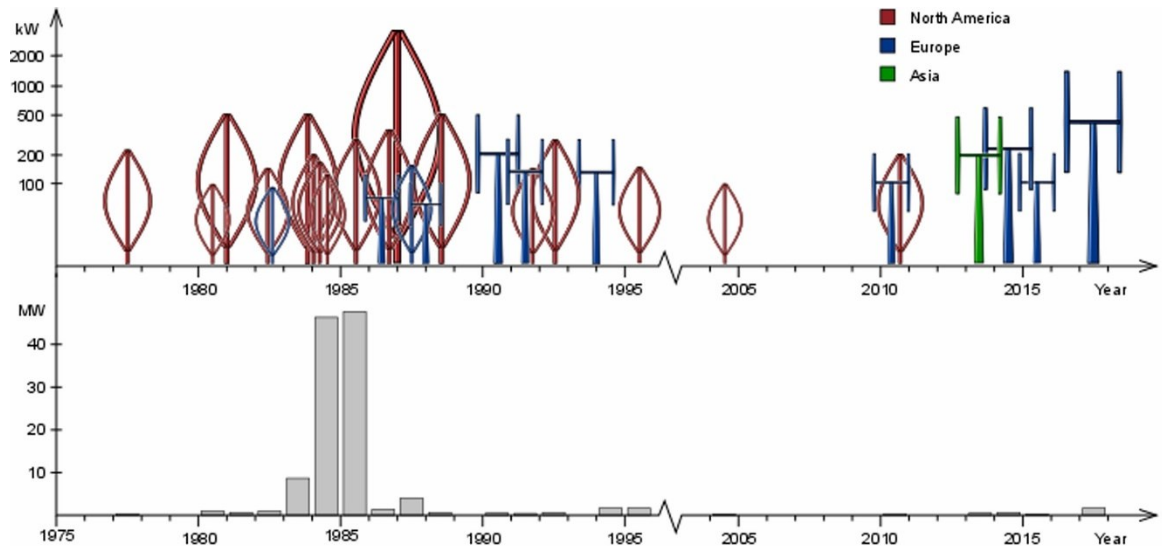


Figure 1.3: Renewed interest VAWT research for turbines over 100kW, specifically in H-Darrieus style VAWT (Möllerström et al., 2019).

1.2. Wind Energy Contributions Towards Energy Generation

Wind energy is a rapidly growing market across the globe with an effort to produce more carbon free energy. The World Energy Outlook (WEO) for 2021 presented by the International Energy Agency projects that electrical energy demand will increase up to 30% by 2030 and 80% by 2050 from the levels in 2021 (Energy Agency, 2021). The WEO analyzes the demand based on many generation sources with natural gas, coal, oil, nuclear, wind, hydro, and solar being the primary contributors. Future projections are largely dependent on international and local policy including stated, announced, pledged and net zero emission scenarios. STEPS is the stated policies scenario and is the most conservative of the four, focusing on actual progress and does not assume announced goals will be reached. APS or announced pledges scenario considers all global climate commitments and that goals will be met. SDS is the sustainable development scenario, based on the Paris Agreement with all pledges and policies reached in full. NZE is the Net Zero Emissions by 2050 scenario and is the most liberal of the four, with a reliance on the global energy sector reaching net zero CO_2 emissions by 2050. Based on these various scenarios, the WEO estimates that renewables (solar, wind, hydro, marine, geothermal and bioenergy) generation shares will be between 40-60% with 23-40% of the contributions coming from wind and solar PV by 2030 and generation increasing to approximately 60-89% with wind and solar contributing to 40-68% in 2050 (Figure 1.4) (Energy Agency, 2021).

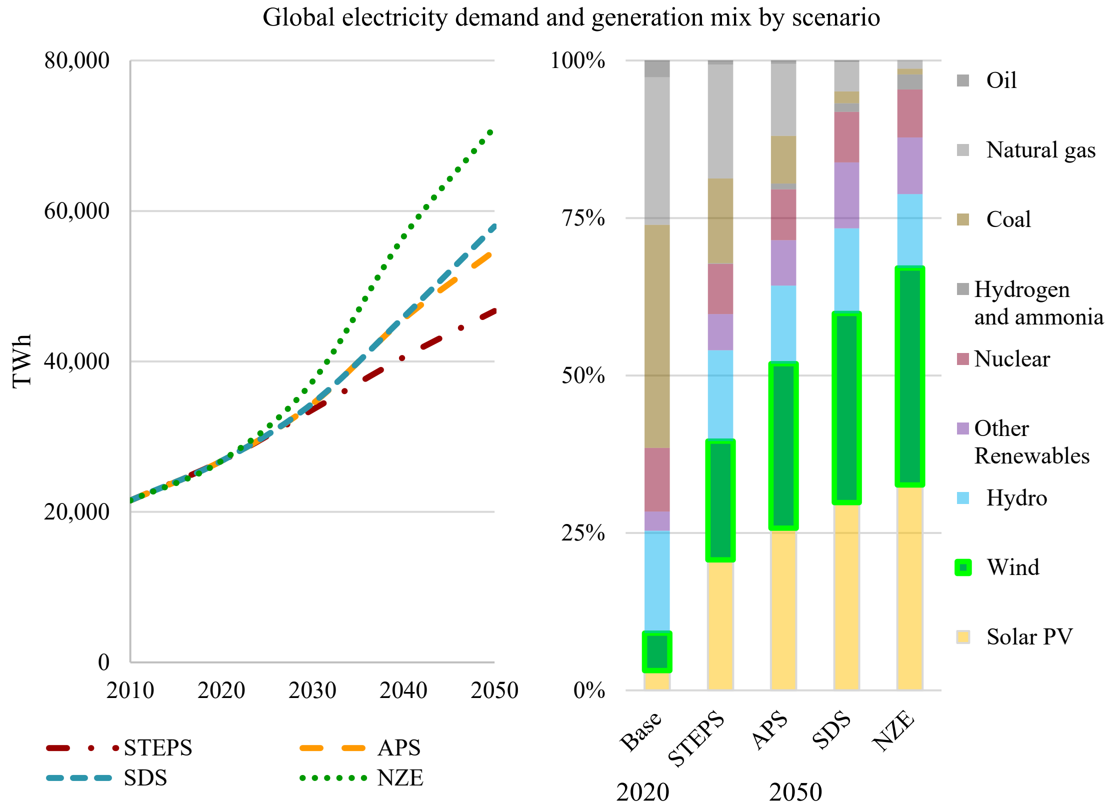


Figure 1.4: Global electricity demand up to 80% increase by 2050 (left) and generation mix with wind contributions highlighted (right) by scenario as a modified presentation from the IEA WEO 2021 report (Energy Agency, 2021).

In the United States, the share of renewables is predicted to double from 2021 to 2050, accounting for more than 57% of generation capacity, based on the Annual Energy Outlook (AEO) from 2022 presented by the Energy Information Administration (EIA) (Energy Information Administration, 2022). Wind and solar are the predominate renewables in the United States as well at 10% and 47%, respectively, with much of the wind share in the next few years based on tax credits (Energy Information Administration, 2022). Natural gas is expected to hold the next majority stake with about 39% of generating capacity, with all other generating technologies accounting for the remaining 4% (Energy Information Administration, 2022).

Figure 1.5 shows the reference case projections from the AEO 2022 report, which is likely to be a conservative estimate with the policies being worked on by the current administration. To achieve a larger contribution of overall generation, there continues to be a need to explore new wind energy technologies, identify supply chain and efficiency gaps, and implement multi-scale distributed wind (and other) energy sources at various integration points through the Nation's energy infrastructure.

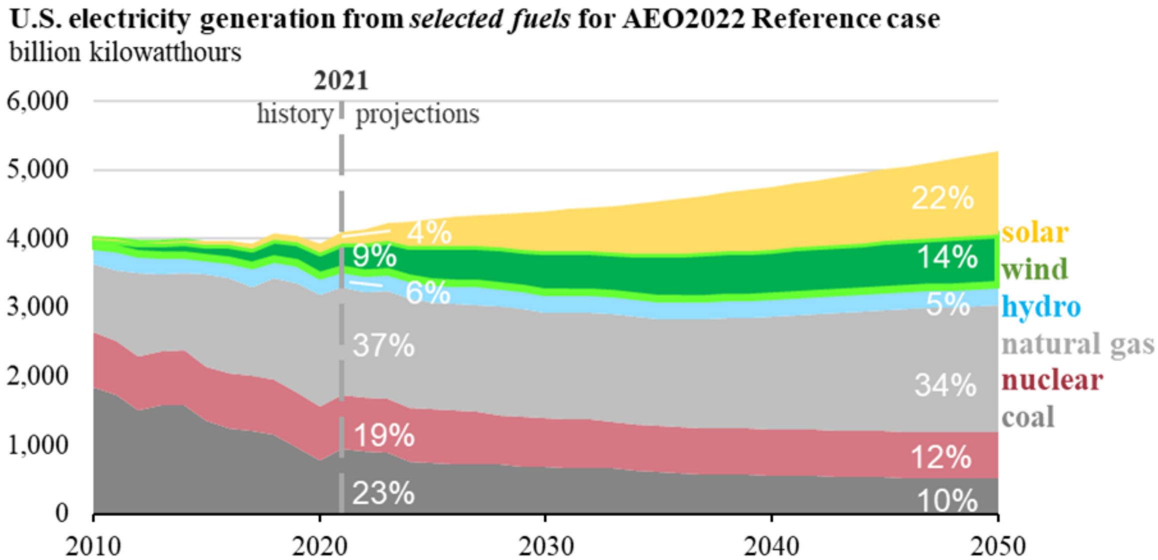


Figure 1.5: Renewables generation shares from selected fuels as a projection from current levels to 2050 with the standard reference case, modified from the EIA AEO2022 Narrative (Energy Information Administration, 2022).

1.3. Types of Wind Turbines

Wind energy has been prevalent since ancient times, being used to pump water or grind grains. Its earliest mention in literature comes from East Asia, with some believing that the Chinese invented windmills over 2000 years ago (Shepherd, 1990). Surprisingly, these early models of windmills (turbines) were of today's less common vertical axis type. Until recently, development of wind turbines had been rather slow, which can be observed when driving through the Midwest region of the United States where many old windmills used to pump water in rural areas are still visible. As previously seen in Figure 1.3, it was not until the 1980s that VAWTs were studied more.

With technological developments over the last century, windmills have experienced major advancements from a simple mechanical system to an intricate electro-mechanical system (i.e., a wind turbine, no longer a windmill). Today's wind turbines capture the kinetic energy of the wind and convert it to electrical energy with a generator. While it sounds simple, these turbines are designed with efficiency and resiliency in mind with complex aerodynamics, specialized materials, adaptive control systems and more unique engineering features to increase efficiency and extend the operational lifespan.

Modern wind turbines are designed to operate at peak performance based on their geographical location and wind resource availability. Wind turbines have a characteristic power curve (Figure 1.6) which categorizes the three primary operating regions of a modern wind turbine. Region 1 is where the turbine is not producing power due to low wind speeds. When the cut-in wind speed is reached (i.e., starting wind speed), the turbine enters Region 2. Most of the operational time is spent in this region until it reaches the rated wind speed. The rated wind speed is where the turbine generator produces the maximum possible rated power for that specific generator. In Region 3, the maximum power is limited and maintained using a variety of control strategies until the turbine cut-out wind speed is reached, which is typically where the turbine shuts down for safety concerns in extreme wind conditions. This region in extreme wind conditions is often called Region 4.

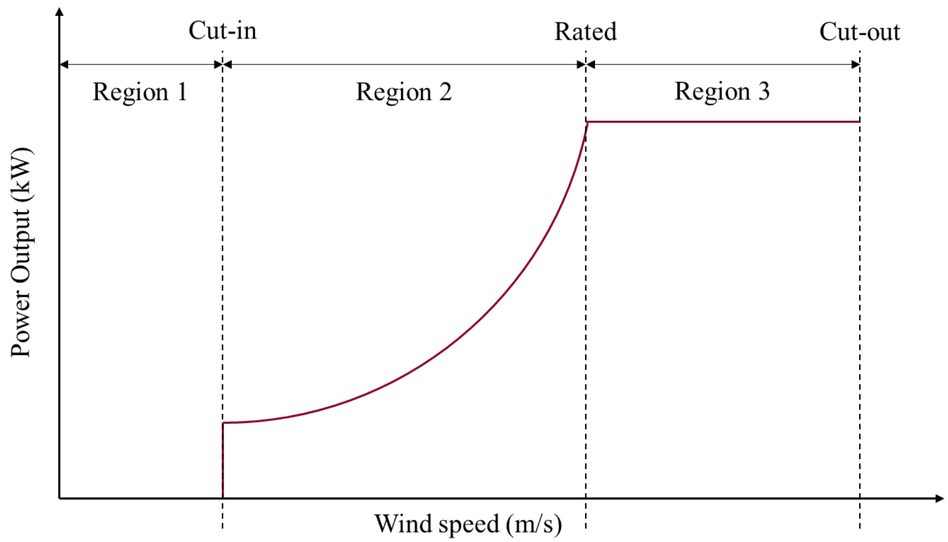


Figure 1.6: Characteristic power curve of a modern wind turbine highlight the three operating regions and key wind speeds.

There are two primary types of wind turbines: horizontal axis (axial flow turbines, or HAWTs) and vertical axis (cross flow turbines, or VAWTs), with the former being widely adopted by the onshore and offshore wind energy industry. As the names imply, the primary difference between the two is the axis of rotation. A HAWT has an axis of rotation that is parallel to the approaching fluid direction, whereas a VAWT has an axis of rotation that is perpendicular to the approaching fluid direction. Within these two categories of wind turbines, there are many different styles at several different scales: small (residential), medium (community) and large (commercial). Small scale wind turbines typically are rated for under 100kW, medium turbines are rated between 100kW and 1MW, and finally large scale wind turbines typically rated greater than 1MW (Tummala et al., 2016). All sizes of VAWTs can be used as distributed wind generators but are currently more common as residential and community scale turbines in these applications. Large scale turbines are most commonly found in commercial energy generation and transmission.

Table 1.1: Scales of wind turbines with rating and use, modified from (Tummala et al., 2016).

Scale	**Small**	**Medium**	**Large**
Rating	< 100kW	100kW - 1MW	>1MW
Use	Residential	Community	Commercial

The creative space in wind energy design is within smaller scale wind turbines, where unique rotor designs have been developed for both HAWTs and VAWTs. Many medium and large-scale turbines are the traditional three-bladed HAWT, largely due to higher power coefficient, C_p, with a moderate tip speed ratio (TSR or λ), as seen in Figure 1.7. The unique rotor designs stem from the fundamental aerodynamics of the turbines drive. The driving wind force results from either drag or lift forces on the blades, which provides the main mechanism for creating torque on the shaft of the generator (Figure 1.8 and Figure 1.9).

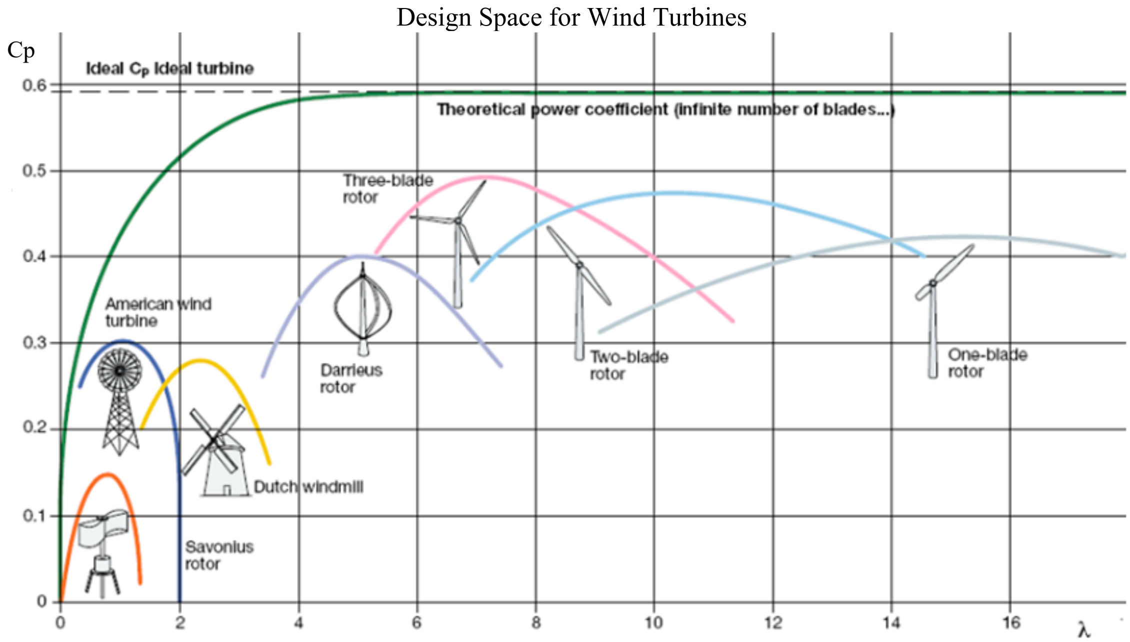


Figure 1.7: Characterization of different types of wind turbines, illustrating the power coefficient, Cp, vs. the tip speed ratio (λ). (Alabdali et al., 2020).

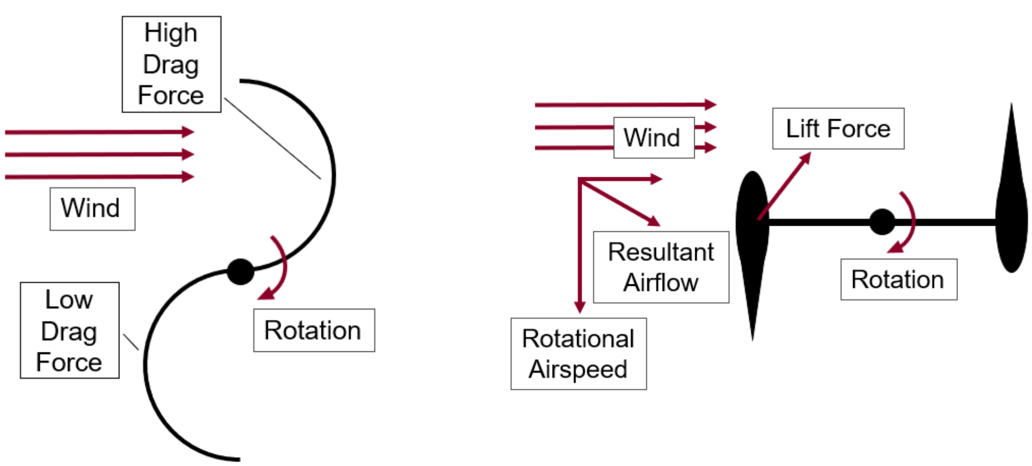


Figure 1.8: General operating characteristics of a drag driven Savonius VAWT (left) and lift driven Darrieus VAWT (right).

Figure 1.9: Lift driven HAWT by S.Tegen/NREL photo 50000 (top left), lift driven VAWT by W. Slocum/NREL photo 65340 (top right), drag driven HAWT by W. Gretz/NREL photo 08199 (bottom left), drag driven VAWT from NW Wind & Solar (bottom right).

VAWTs have several advantages compared to HAWTs due to the simplicity of their design, scalability and offshore suitability (Hand et al., 2021). The typical systems of a wind turbine include the rotor, power train, support structure, foundation, and ground level equipment. The arrangement of these systems can vary with the size (residential, community and commercial scale) and style (HAWT and VAWT) of a wind turbine. When comparing a HAWT to a VAWT costs can be lower for a VAWT in part to not needing a yaw system and active blade pitch mechanisms (Hand et al., 2021). The current levelized cost of energy (LCOE) of floating offshore wind is approximately $200/MWh (2021) (Musial et al., 2022), while a VAWT is has a baseline LCOE of $274/MWh (2018) (Ennis & Griffith, 2018). With development in technology, the VAWT LCOE is predicted to come down to $110/MWh, with the overall offshore wind LCOE to drop between $58-$120/MWh (Ennis & Griffith, 2018; Musial et al., 2022). Floating offshore wind is likely where the most VAWT development will take place due to a sea-level generator which is estimated to reduce combine operations and maintenance cost by around 25% when compared to an equally sized HAWT (Ennis & Griffith, 2018).

VAWTs fall into three main categories: Savonius, Darrieus and H-Darrieus rotors based on their drive type and physical characteristics (Figure 1.10). Savonius rotors are drag driven with a low maximum power coefficient (Cp ~ 0.15) and tip speed ratio (TSR) ($\lambda \sim 0.85$). The rotor on a Savonius "catches" the incident wind creating a drag force, i.e., a pressure differential between the airfoil surfaces, due to the convex side having a lower drag than the concave side. The relatively poor performance is due to the drag driven mechanics, limiting it to rotate at the speed of the incident wind (Óskarsdóttir, 2014). While the performance of a Savonius rotor is poor, it can benefit in simplicity. A Savonius rotor can easily be assembled and used in low power situations. They are occasionally paired with a Darrieus rotor to help with cut-in at low wind speeds. Darrieus rotors are lift driven, as depicted in Figure 1.8 and are more efficient than Savonius rotors. They operate similar to an airplane wing and HAWT turbine blades, creating a high-pressure region on one side of the airfoil and a low-pressure region on the other side resulting in a lift force, creating torque, and driving the rotor around the central axis. A typical Darrieus rotor has an "egg beater" like shape, more formally known as a troposkein (Óskarsdóttir, 2014). This shape acts like a wire in tension, reducing the overall forces across the blade. Blades (airfoils) of this shape are difficult to manufacture on a large scale, which leads to the more practical style of the H-Darrieus rotor. H-Darrieus rotors have similar mechanics to the Darrieus rotor, in that they are lift driven but differ in the airfoil shape. The H-Darrieus rotor forgoes the troposkein shape with the blades being straight. The straight blades are cheaper and easier to manufacture making this style more practical and easier to scale. The loading on the blades is variable with the changing angle of attack and the resulting centrifugal force. Helical blades can be implemented to evenly distribute the torque over a full revolution of the turbine, reducing the destructive forces on the turbine. More of these characteristics will be discussed in the coming section.

Savonius | Darrieus | H-Darrieus

Figure 1.10: High-level depictions of VAWT categories with a Savonius by from NW Wind & Solar (left), a Darrieus by Arborwind (middle) and an H-Darrieus by UMD (right).

1.4. Wind Turbine Performance

Wind turbines convert the kinetic energy of the wind into electrical energy. The power available in the wind (Eqn. 1) is described by,

$$P_{wind} = \frac{1}{2}\rho A U_{\infty}^{3}$$

(Eqn. 1)

where ρ is the air density [kg/m^3], A is the rotor swept area [m^2] and U_∞ is the free stream velocity [m/s] of the wind typically measured at the center of the rotor. In the case of an H-Darrieus VAWT, the swept area is calculated as the diameter of the rotor times the length of the blade (Möllerström et al., 2016). When evaluating the performance of a wind turbine in a variable climate such as Duluth, MN, the air density can play a significant role. Based on the historical observations for the record high 41.1°C (106°F) and low -40.6°C (-41°F) temperatures at the Duluth International Airport, there can be up to a 26.0% change in available power at similar wind speeds when using standard atmospheric pressure (1013.25 millibars). This can be further explained by the ideal gas law and mass transfer. The power produced (Eqn. 2) by the turbine is defined by,

$$P_{turbine} = T\omega$$

(Eqn. 2)

where T is the shaft torque [Nm] produced by the turbine and ω is the rotational speed [rad/s]. The ratio of the power produced by the turbine and the available power in the wind defines the power coefficient, C_p, given by equation (Eqn. 3),

$$C_p = \frac{P_{turbine}}{P_{wind}} = \frac{T\omega}{\frac{1}{2}\rho A U_\infty^3}$$

(Eqn. 3)

The power coefficient is representative of the aerodynamic efficiency of the wind turbine, and often is reported in relation to the tip speed ratio, TSR, or otherwise noted as λ defined in equation (Eqn. 4),

$$TSR = \lambda = \frac{\omega R}{U_\infty}$$

(Eqn. 4)

where R is the radius [m] of the turbine rotor. Given that a wind turbine does not completely stop the wind (i.e. remove all kinetic energy from the wind), physicist Albert Betz derived that the maximum theoretical C_p of an un-ducted wind turbine is 59.3%, better known as the Betz Limit (Kelvin Lwangu, 2020).

1.5. VAWT Characteristics

VAWTs have developed a renewed interest in the wind energy research, development, and deployment community. H-Darrieus turbines have a lower cost when compared to other types of turbines and their ability to be configured in different arrays to increase performance has driven this renewed interest. Analysis with computational and experimental methods have evolved for VAWTs, providing a better understanding of the aerodynamics, fluid-structure interactions, and performance characteristics. To better compare VAWT performance characteristics throughout the industry Barnes et al, (2021) has proposed a standardized method, which uses *Cp*, TSR, aspect ratio and blade density/chord length, after analyzing numerous research campaigns. Some other turbine design characteristics include solidity, strut design, leading/trailing edge perforations and aspect ratio, among many others (Barnes et al., 2021).

In the VAWT research area there have been many CFD simulations (Barnes et al., 2021; Miliket et al., 2022, among others), numerous wind tunnel experiments are available (Kelvin Lwangu, 2020; Miller et al., 2021, among others), yet there is a general lack of field scale studies. It has been found that there are discrepancies between wind tunnel and field scale studies leaving a knowledge gap and thus further limiting the development of VAWTs. When comparing *Cp* vs. TSR curves between wind tunnel experiments and field studies, wind tunnel experiments typically show a higher *Cp* at a lower TSR (Barnes et al., 2021). This discrepancy can be attributed to differences in the chord-based Reynolds number, which is the ratio of the relative velocity (U_r) times the chord length of the airfoil (c) divided by the fluid kinematic viscosity (ν) given by (Eqn. 5)

$$Re_c = \frac{U_r c}{\nu}$$

(Eqn. 5)

which has a great impact on the performance of a VAWT. In true VAWT operations, the chord-based Reynolds number uses the relative velocity at the blade, which is the resultant of both free-stream velocity and the local blade speed. For a VAWT, this chord-based Reynolds number can vary greatly during a single rotation while the blades rotate a full 360 degrees. This has been studied in depth and the literature agrees on the Reynolds number dependency, with most small turbines with rotors diameters of only several meters, operating at $Re_c < 10^5$. Achieving a $Re_c > 10^6$ will provide better scalability to commercially sized wind turbines (Barnes et al., 2021; Bogateanu et al., 2014; Hand et al., 2021). The NACA 0018 airfoil is common throughout VAWT studies and highlights the Reynolds dependency well in Figure 1.11. For low TSR, a Darrieus wind turbine *Cp* is largely dependent on the Reynolds Number. When approaching a higher TSR around 3.5, the *Cp* is less dependent on the Reynolds Number when it approaches $Re \approx 5\text{x}10^5$.

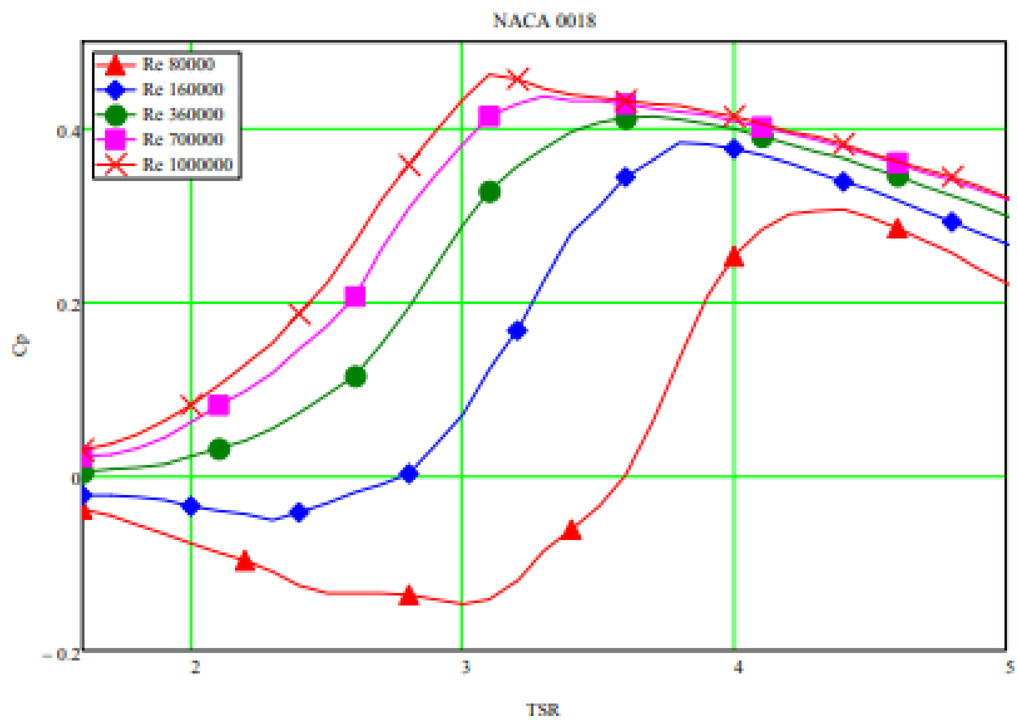


Figure 1.11: Coefficient of Performance, Cp, across a range of tip speed ratios (TSR ≈ 1.7 to 5) highlighting the Reynolds number dependency on the performance for a NACA 0018 airfoil during wind tunnel tests of a VAWT (c/R=0.083) from (Bogateanu et al., 2014).

1.6. Aerodynamics

Omnidirectionality is one of the VAWTs most important characteristics, allowing them to operate with winds approaching from any direction and forgoing any complex yaw mechanisms. This also leads to a unique dynamic stall characteristic that results from a rapidly changing local angle of attack (AoA or α) experienced by the blades sweeping through the full turbine regime (-180° to 180°) on each rotation (Figure 1.12). Dynamic stall can have grave consequences on a turbine, increasing the noise, lowering efficiency, and inducing vibrations. When rotating, the relative velocity changes as a function of the freestream velocity, U_∞, the rotational velocity, $U_b = \omega R$, and the azimuth angle, θ, of the rotor (Figure 1.12)(Eqn. 6) (Barnes et al., 2021).

$$U_r = U_\infty \sqrt{\left(\frac{\omega R}{U_\infty} - \cos(\theta)\right) + (sin\theta)^2}$$

(Eqn. 6)

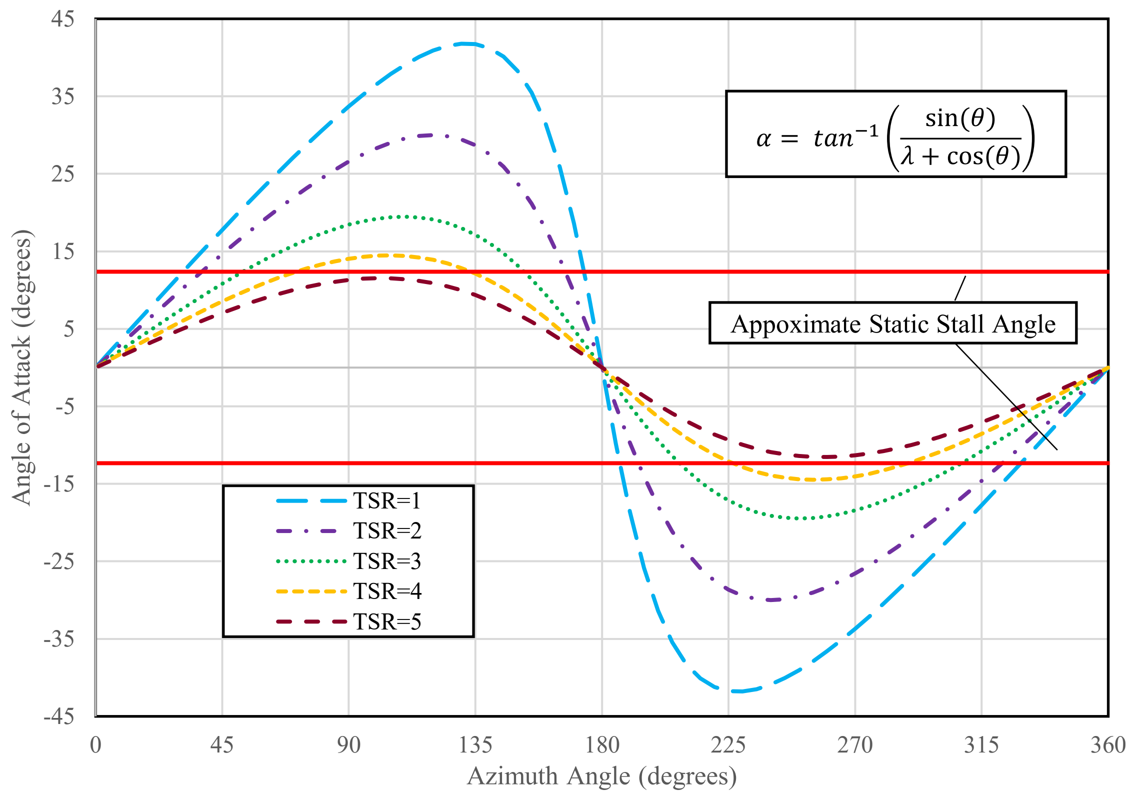


Figure 1.12: Theoretical instantaneous angle of attack variation for a H-Darrieus VAWT with an approximate static stall angle of 13 degrees, recreated based on (Pearson, 2013).

The AoA *(α)* is given as the angle between the blade chord and the relative wind speed U_r, given by (Eqn. 7) and depicted below (Figure 1.13),

$$\alpha = tan^{-1}\left(\frac{\sin(\theta)}{\lambda + \cos(\theta)}\right)$$

(Eqn. 7)

which is representative of the instantaneous AoA under an assumption of a uniform freestream flow. The local AoA is found using the induction factor, which accounts for the power extracted by the rotor (Gosselin et al., 2013; Sagharichi et al., 2018). The power of the turbine can be found with equation (Eqn. 2), which is related to the blade torque as $T = F_T * R$, where F_T is the tangential force. The tangential force can be found by decomposing the lift *(L)* and drag *(D)* forces to find the resultant force *(F_R)*. After finding the resultant force *(F_R)*, it can be further decomposed to find the normal force *(F_N)* and tangential force *(F_T)*, which is further depicted in Figure 1.13 (Carrigan et al., 2012).

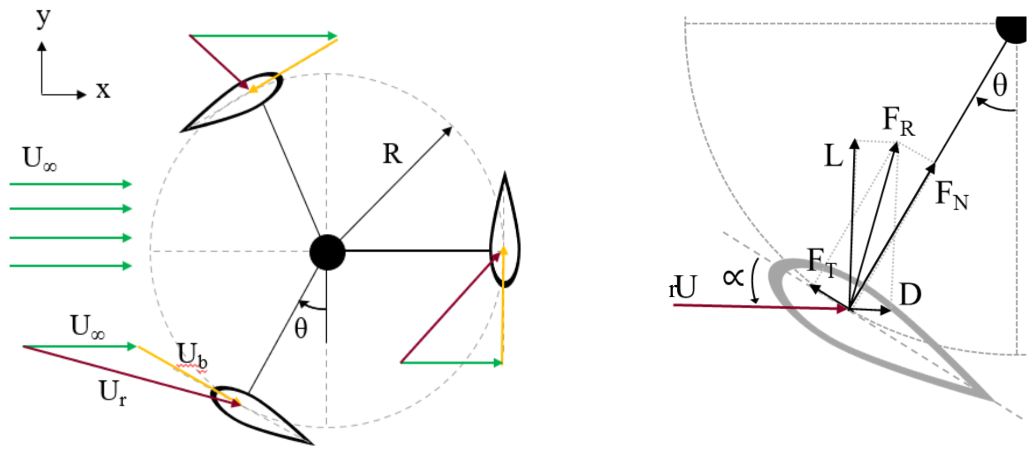


Figure 1.13: Relative velocity at various locations of a VAWT and the resulting AoA (left) and the aerodynamic decomposition of the torque driving force FT (right) reproduced from (Keijer, 2020).

Dynamic stall is the result of a five-step process (Figure 1.14), outlined by McIntosh, in which the flow separates from the airfoil and reverses to travel from the trailing edge to the leading edge of the airfoil across what was originally the low-pressure side. As the AoA increases, vortices are created resulting in reduced lift production. This flow eventually fully separates from the airfoil surface, and a sharp decrease in lift occurs until the AoA falls well below the static stall angle on the upwind stroke (McIntosh, 2009; Pearson, 2013). The flow separation is further depicted using a CFD studying a hydrokinetic cross flow turbine, showing how the azimuth angle effect the vorticities (Figure 1.15).

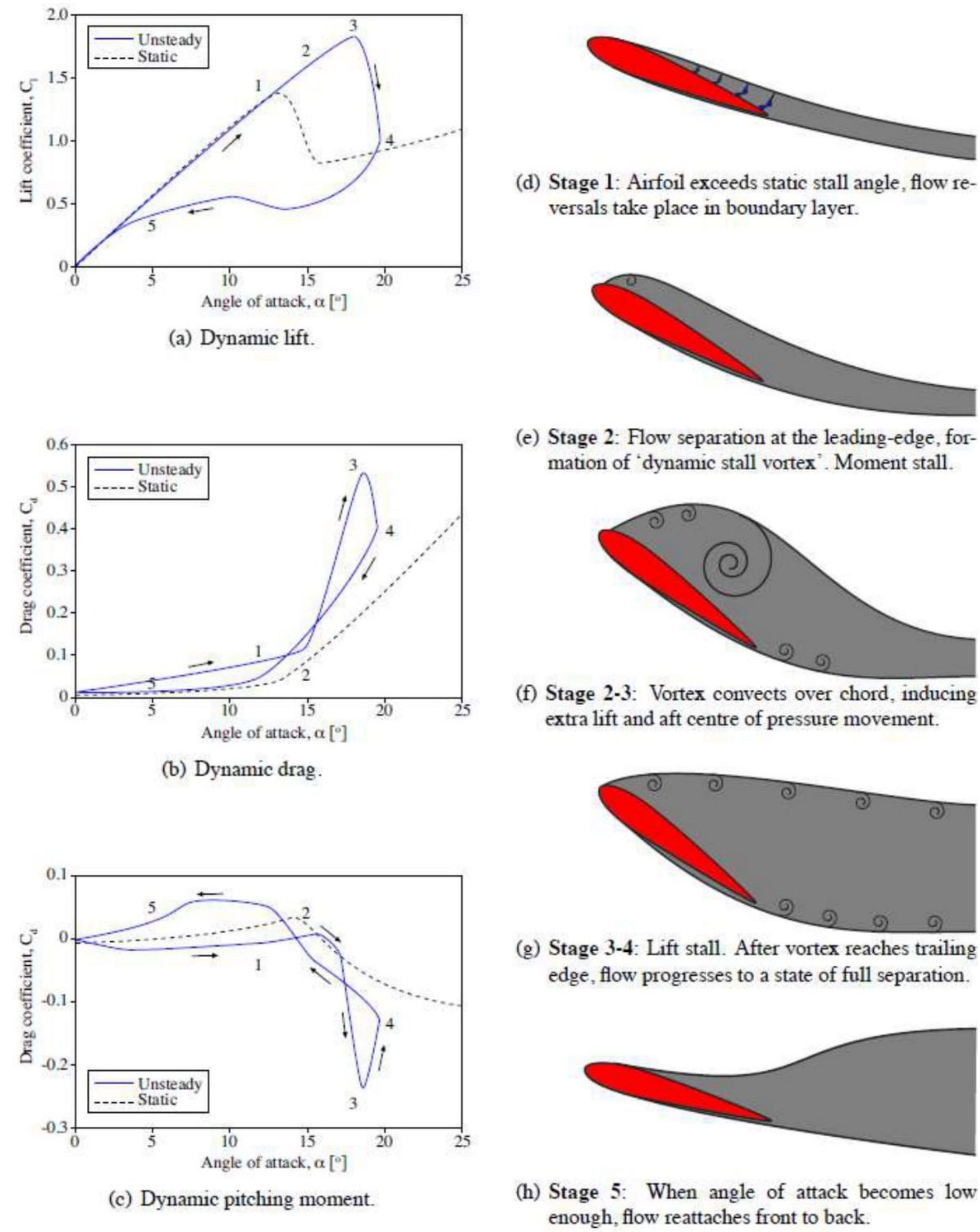
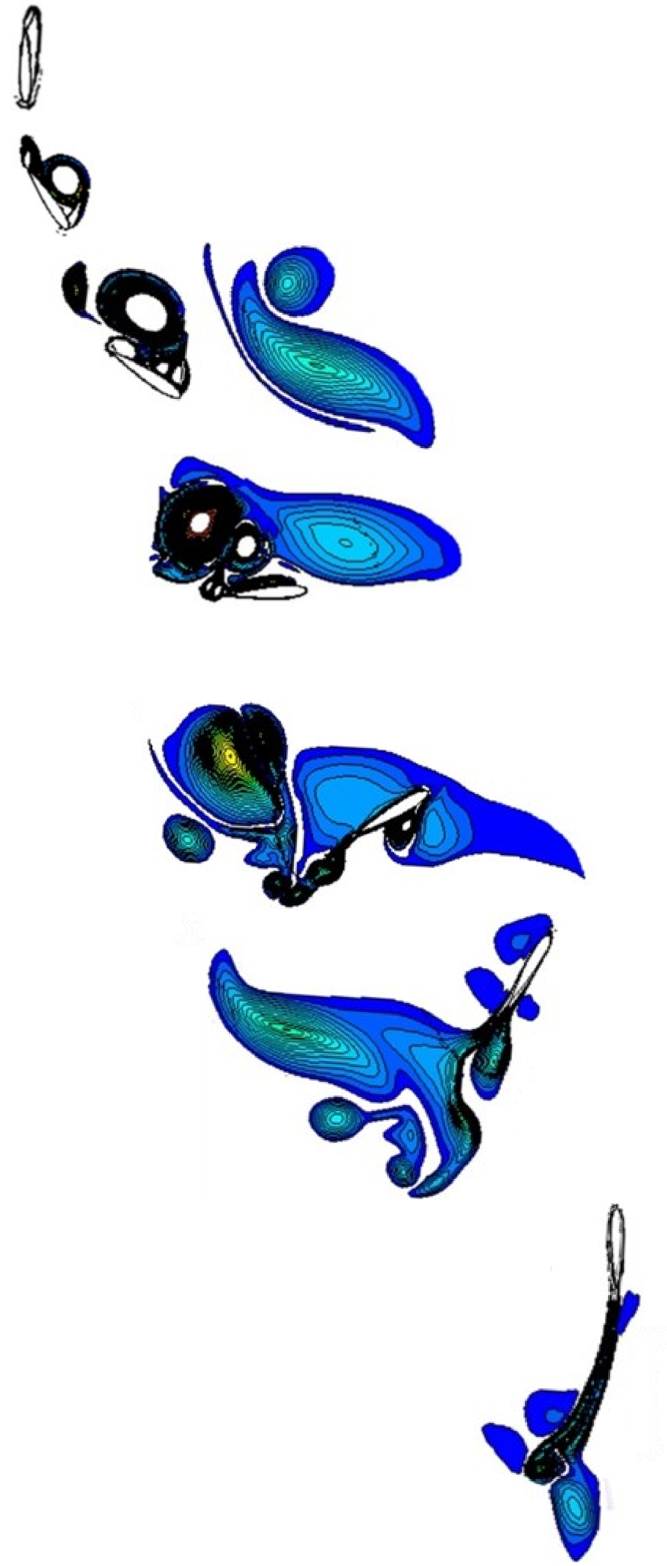


Figure 1.14: Five-step (d-h) process of dynamic stall due to the changing angle of attack (0-25°) of a VAWT blade and the corresponding dynamic lift(a), drag(b) and pitching moment(c) throughout the cycle (McIntosh, 2009).



α=90°: Flow is attached to the profile, although a small vortex is started on the low-pressure side. No vortex is shed from the blade.

α=120°: Vortex grows bigger and is about to detach from the profile.

α=150°: Vortex expansion on the low-pressure zone takes place as the blade heads to hit the vortex released by the preceding blade.

α=180°: Complexity in the flow characteristics due to large turbulence scales increases as there is a vortex-vortex interaction. At higher tip-speed ratios, these strong unsteady vortex structures diffuse slowly compared to the turbine's rotation and catch up to the following blade.

α=210°: Blade penetrates the vortex which results in an impulsive change in the blade's hydrodynamic loading.

α=240°: Blade-vortex interaction is ending, resulting vortex is left behind the blade. Progressive reattachment of the flow over the blade is evident. Another weak vortex detaches from the leading edge, which does not survive for a long time.

α=270°: Flow is completely reattached with negligible perturbations.

Figure 1.15: Hydrokinetic CFD study showing the vortex development with a progressing azimuth angle (α) modified from (Gorle et al., 2019).

With variable AoA and the dynamic stall phenomenon comes the investigation of fluid structure interactions with the blade and the effect on performance. Conceptually, vorticity *(Ω)* is the tendency for a fluid participle to spin. Mathematically it is given as the curl of the velocity field, $\Omega = \nabla \times u$ which can be simplified with the vorticity transport equation, with the vortex stretching ($(\omega \cdot \nabla)u$) and diffusive ($v\nabla^2\Omega$) terms as,

$$\frac{D\Omega}{Dt} = (\omega \cdot \nabla)u + v\nabla^2\Omega$$

which relates to the circulation *(Γ),*

$$\Gamma = -\oint_C \vec{U} \cdot d\vec{l} = \Omega \iint_s \vec{\Omega}\, dS$$

The circulation can be physically thought of as the force which pushes a fluid particle around a path and can be correlated to the lift force produced on a VAWT blade (Keijer, 2020). The relationship of circulation is given as,

$$L = \rho\Gamma|U_r|$$

noting that Γ in this case is the circulation bound to the turbine blade and U_r is the relative velocity at the blade using the two-dimensional vortex model (Wilhelm, 2010).

Circulation and vorticity have been investigated using computational models such as CFD and has a fair amount of particle image velocimetry (PIV) studies completed using small scale wind tunnel experiments. Sun and Zhou (2021) reviewed literature regarding the vorticity around H-Darrieus VAWTs, specifically the comparison between CFD and PIV models. One notable finding is that the CFD techniques typically provide a good representation of the of vorticity fields when compared to PIV. Figure 1.16 represents these findings using an Unsteady Reynolds-averaged Navier-Stoke (URANS) CFD simulation and a PIV experiment, from which we can see the TSR has a great effect on the vorticity, resulting in a lager vortex at a lower TSR. Throughout other reviews completed by Sun and Zhou, they determined that shear stress transport (SST) k-ω accurately captured the vortex generation contributed to dynamic stall (Sun & Zhou, 2021). However, these studies focus on small scale devices and lower Reynolds numbers not reflective of field-scale conditions.

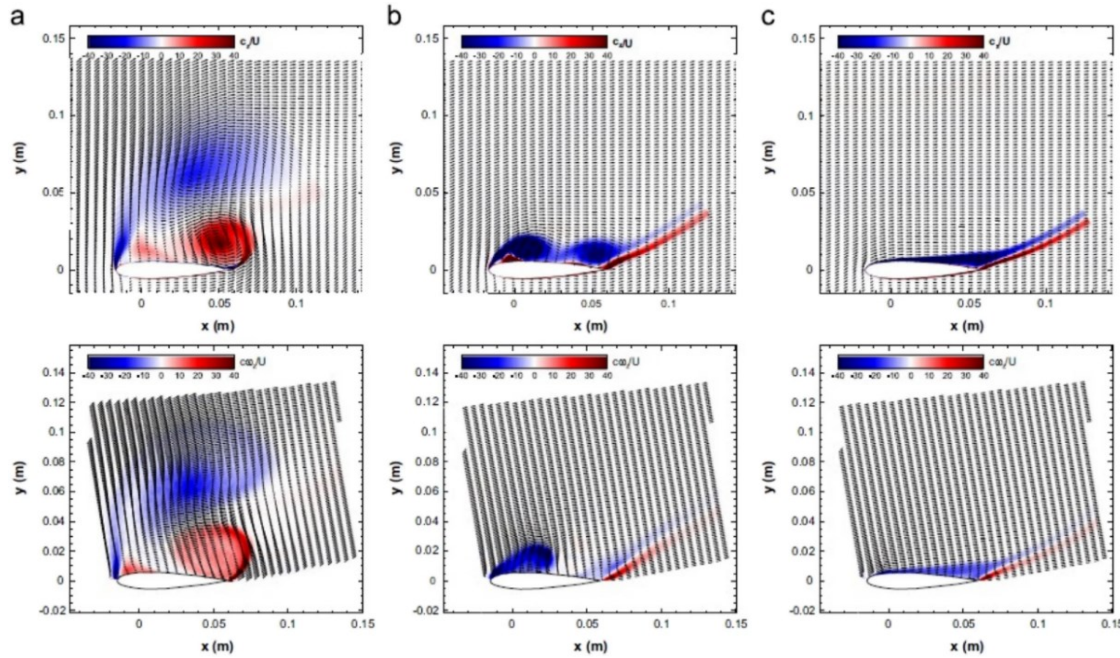
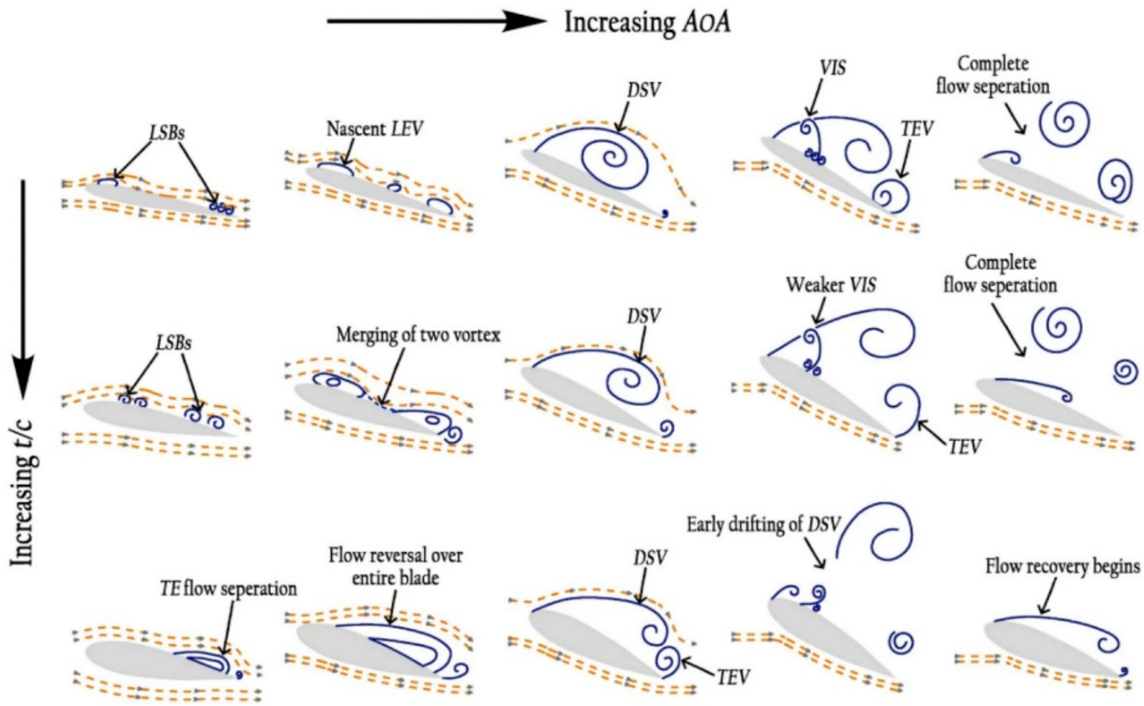


Figure 1.16: URANS CFD simulation (top) compared to PIV experiment (bottom) for a H-Darrieus VAWT with a TSR of 1 (a), 2 (b), 3 (c) (Sun & Zhou, 2021).

Another interesting finding in the review done by Sun and Zhou is the relationship between an increasing AoA and the thickness to chord ratio (t/c), where the thickness is the maximum thickness of the blade, and the chord is the straight-line distance from the leading edge to trailing edge. The primary differences being the origin and separation of the vortex with increasing t/c and magnitude of the vortex with increasing AoA as seen in Figure 1.17 (Sun & Zhou, 2021).



LSB =leading edge separation bubble, LEV = leading edge vortex, TEV = trailing edge vortex, DSV = dynamic stall vortex, VIS = vortex-induced separation

Figure 1.17: Stages of dynamic stall when compared by the t/c and AoA (Sun & Zhou, 2021).

A CFD study completed by Kozak showcases the vortex generation at different azimuth angles for a three-bladed H-Darrieus VAWT (Figure 1.18). It can be observed that the vorticity starts at roughly θ=90° and sheds around θ=180° for a TSR=2.0. Other studies demonstrated that the vorticity was much smaller with high TSRs. With a three-bladed VAWT, when one blade is suspect to dynamic stall another blade is generating lift, which helps keep the turbine operational. It is unclear on how the downstream blade interacts with shed vorticities from the upstream blade (Kozak, 2016), indicating a need for exploring higher Reynolds number VAWT characteristics at field-scale, including the vortices inside the rotor that travel downwind and potentially interact with blades on the downwind side of the rotor.

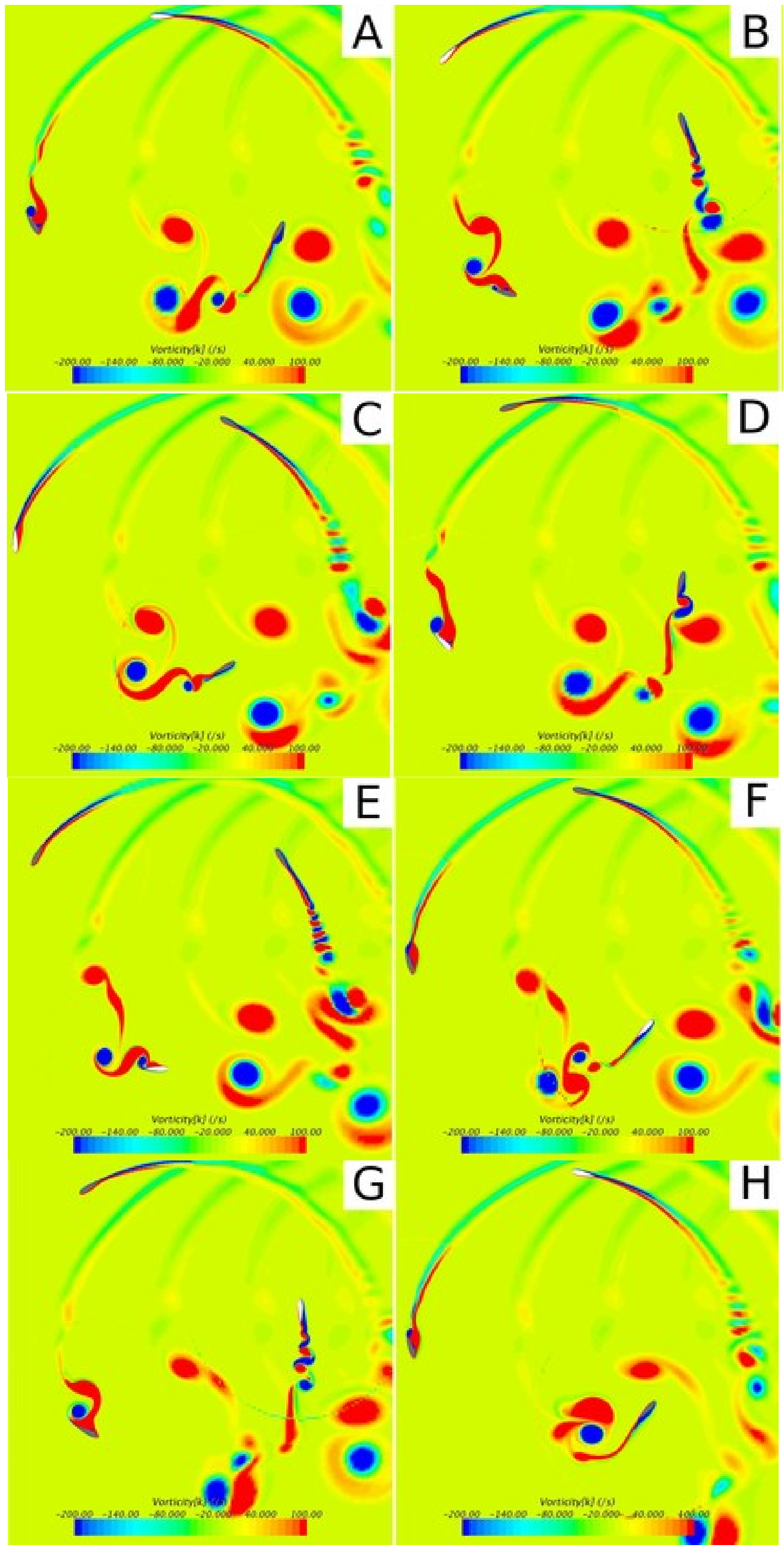


Figure 1.18: Vorticity field for a H-Darrieus VAWT with frames A=0° to H=315° at 45° increments of the azimuth angle and a TSR=2.0 (Kozak, 2016).

1.7. Field Scale VAWT Studies

As previously mentioned, many CFD and wind tunnel studies have been completed, but there is a significant lack of field scale VAWT studies. Wind tunnel experiments are limited on VAWT size and Reynold's dependency, with the larger end of experimental setups having a rotor diameter up to 2.0 meters and $Re_c < 3 \cdot 10^5$. Many of the studies that have been completed focus on the specific characteristics of wind turbines and how they relate to the power coefficient. Some common characteristics investigated include the blade pitch, solidity, and airfoil design and how they affect the TSR and power coefficient. Field scale studies allow for the investigation of turbine characteristics at higher Re, realistic turbulence intensity (TI, intensity of wind velocity fluctuations), temperature influence and other realistic circumstances.

A study completed by Möllerström et al. on a 200kW H-Darrieus VAWT investigated the influence of turbulence on the power produced. The turbine, outlined in Table 1.2 (Figure 1.19) and located in Sweden has since been derated to operate at 80kW due to structural concerns with the wooden tower. The study found that the relationship between power production and turbulence is low and there is a small increase in power in high turbulence. This study also suggested that control systems should be tuned based on the TSR and TI. Initial investigations found a higher TSR during more turbulent conditions resulted in better power performance, with an optimal TSR = 4.3 providing a $C_p = 0.33$ for this turbine.

Figure 1.19: 200kW T1-turbine by Vertical Wind AB and Uppsala University from (Möllerström et al., 2016).

Table 1.2: Specifications for the 200kW T1-turbine by Vertical Wind A and Uppsala University.

Parameter	**Specification**
Rated Power (reduced rating)	200 kW (80 kW)
Rotor Diameter	26 m
Hub Height	41 m
Blade Length	24 m
Configuration	3 blade fixed pitch
Cut-in Wind Speed	4 m/s
Rated Wind Speed	12 m/s

Li et al. completed a study comparing the performance of a small VAWT in a wind tunnel and field scale (Figure 1.20), primarily investigating the power coefficient and TSR relationship and wake recovery. Details of the turbine are provided in Table 1.3. The wind tunnel and field scale studies yielded similar results with a slightly lower Cp=0.19 with increased TSR=2.5 in the field when compared to the wind tunnel at a Cp=0.2 and TSR=2.1, at the respective optimal operation point. The relationship with TI seemed to have little affect at low TSR<1.8, but while the TSR increased (TI>1.8), the turbine performed better with lower TI. This is likely attributed to the relationship between TI and the wind speed and Reynolds number. The TI decreases as the wind speed and Reynolds number increase. The wake recovery was shown to be quicker in the field study with 90% or greater recovery to free stream at 2 rotor diameters from the axis of the turbine. The study also showed that the wake recovery is reduced from 50-75% of the free stream velocity in the downwind advancing region of the turbine while TSR increases from 1.38-2.58 at 1 rotor diameter from the turbine.

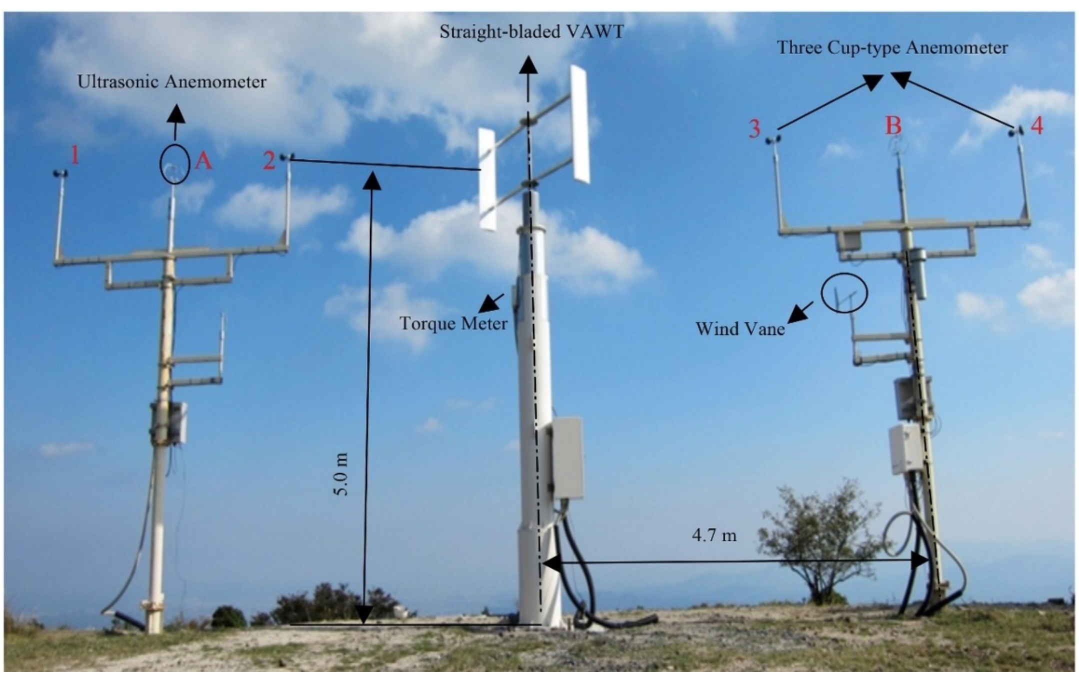


Figure 1.20: Experimental setup used by Li et al. for a small scale VAWT field study (Li et al., 2016).

Table 1.3: Specifications for the small scale VAWT studied by Li et al. using wind tunnel and field setups.

Parameter	**Specification**
Rotor Diameter	2 m
Hub Height	41 m
Blade Length	1.2 m
Configuration	2 blade fixed pitch
Rotational Speed (torque sensor)	0-6000 rpm

California Institute of Technology (Caltech) has a research site in the Northern part of Los Angeles County called the Caltech Field Laboratory for Optimized Wind Energy (FLOWE) where they have been completing research on VAWT arrays of up to 18 turbines (Figure 1.21 and Table 1.4). Studies at FLOWE suggest that VAWT arrays are effective even though there are blockage effects because turbines can experience performance enhancements while downwind from another if the upstream regions align due to flow acceleration in the wake (Brownstein et al., 2016; Kinzel et al., 2015).

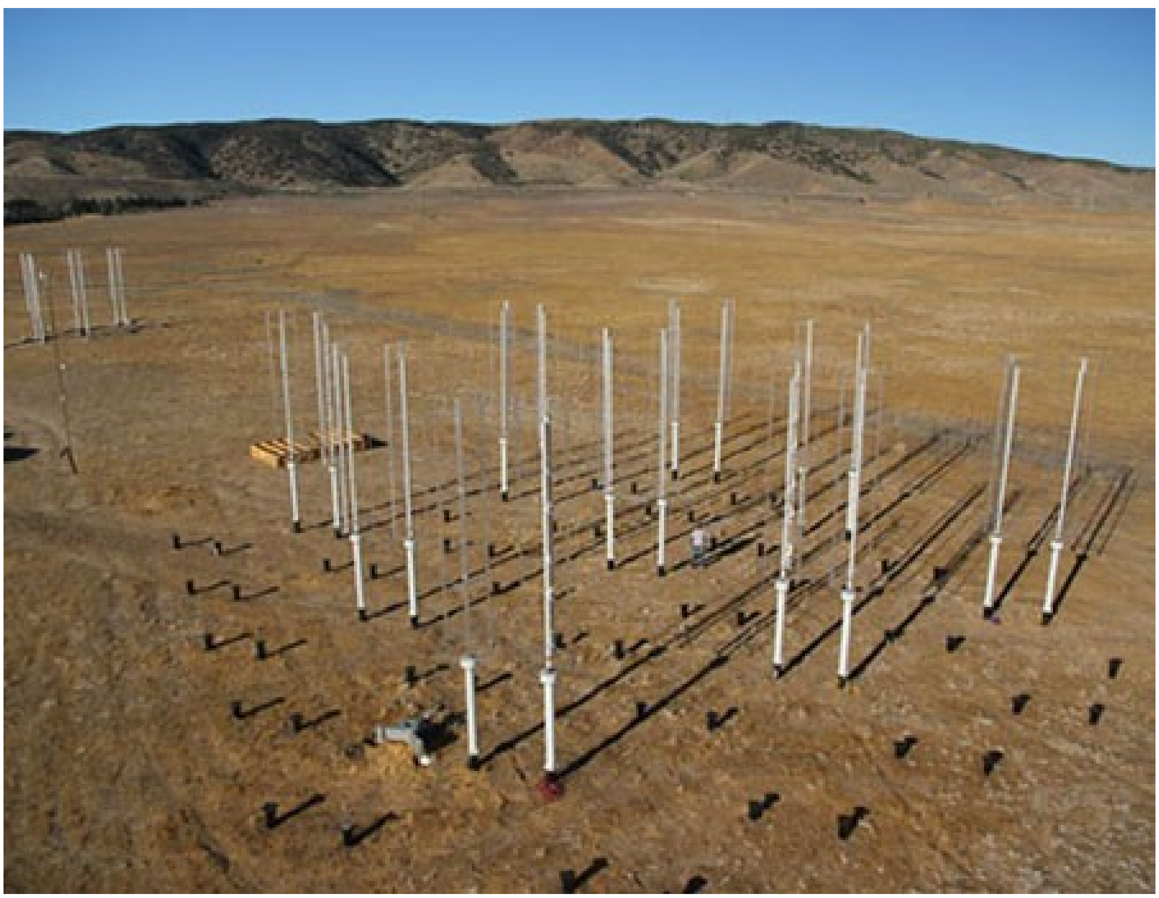


Figure 1.21: Caltech FLOWE research facility featuring an 18 VAWT array (Kinzel et al., 2015).

Table 1.4: Caltech FLOWE VAWT specifications.

Parameter	Specification
Rated Power	1.2 kW
Rotor Diameter	1.2 m
Blade Length	3 m
Configuration	3 blade fixed pitch
Cut-in Wind Speed	3.8 m/s
Rated Wind Speed	12 m/s
Rotational Speed	0-420 rpm

Wei, et al. competed a study at Caltech FLOWE using 3D particle tracking velocimetry of snowflakes (Figure 1.22) to examine the vortex structures of a 2kW VAWT with 5 straight blades and a 1kW VAWT with 3 helical blades. The study shows strong vorticities on the side of the turbine rotating into the wind, as seen in Figure 1.23 (Wei et al., 2019).

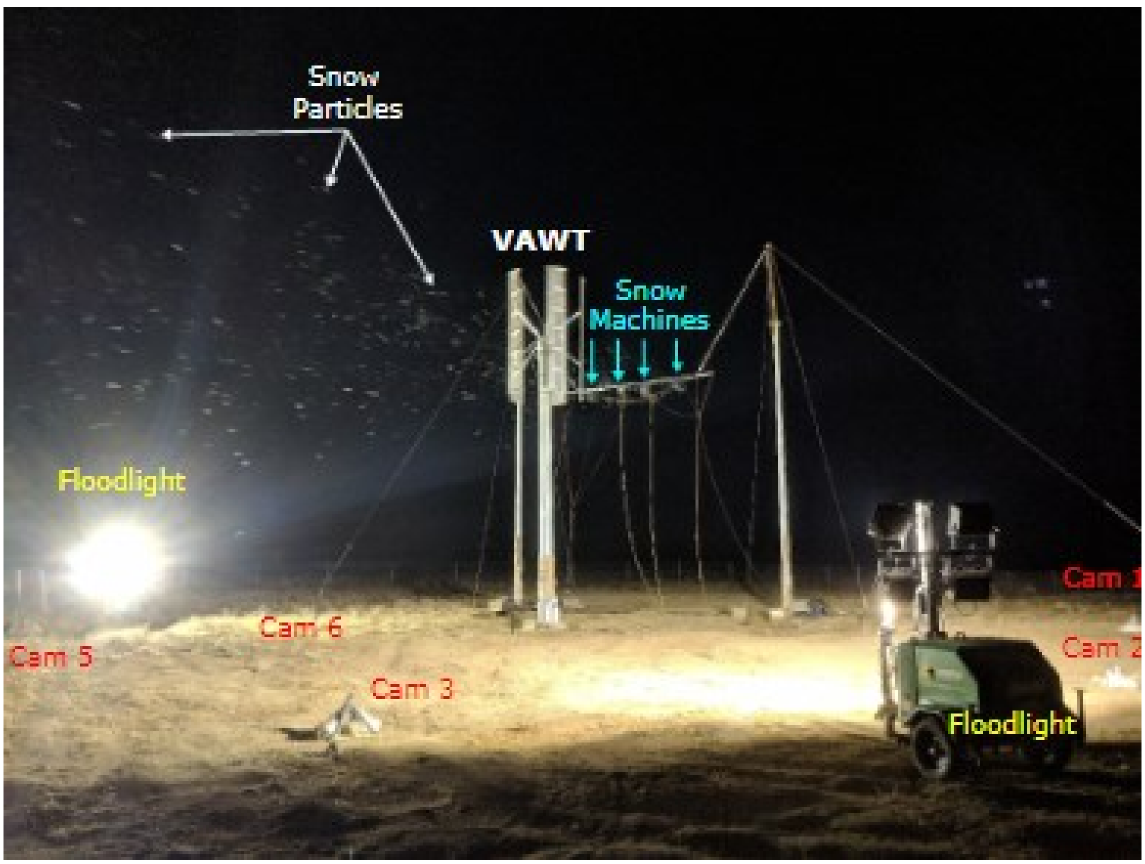


Figure 1.22: PIV field study completed at FLOWE using snow particles and a 5-bladed VAWT (Wei et al., 2019).

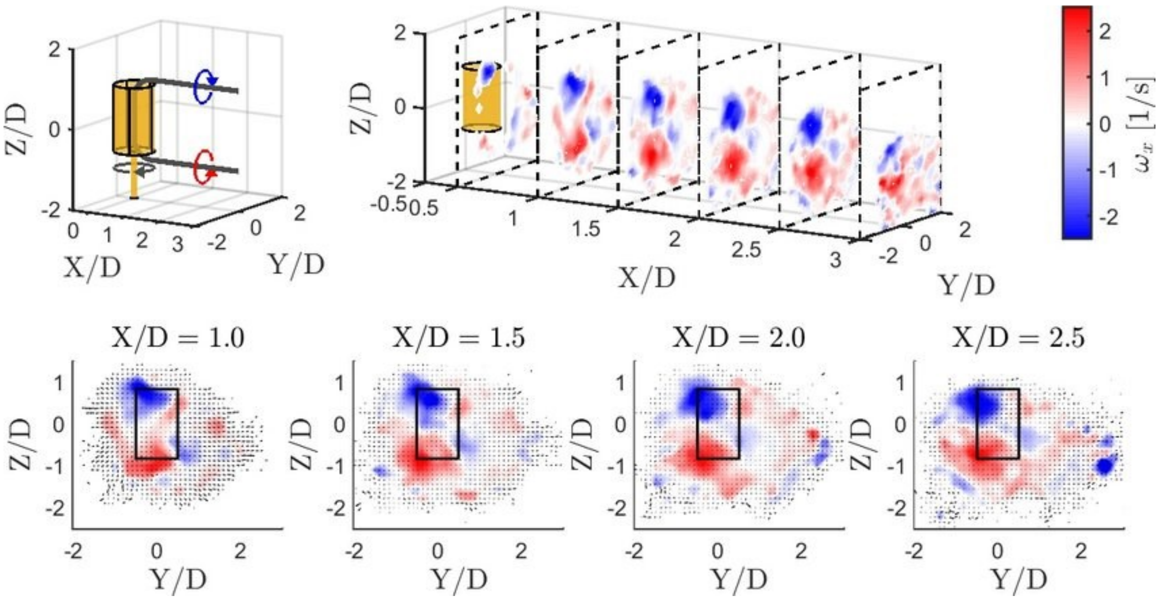


Figure 1.23: 3D visualization of vortex structures formed by a H-Darrieus VAWT with TSR=1.2, showing counter rotating vertical vortices in the turbine wake (Wei et al., 2019).

1.8. Meteorological Observations

Meteorological observations are the primary basis to categorize the wind conditions at a site. These observations assist with critical assessments, such as the wind shear, wind veer and turbulence intensity which are among a few atmospheric conditions that affect the performance of a wind turbine. The standard methods for wind turbine characterization are set forth by the International Electrotechnical Commission (IEC) 61400. These standards are based off a 10-minute averaging periods and have been tested by Goit et al., suggesting that a smaller averaging window of 30 or 60 seconds should be used (Goit et al., 2022). This is since smaller wind turbines (less than 15m rotor diameter) have a shorter time response and the power prediction is more accurate.

Wind shear and wind veer are terms which describe how the wind changes with height above the ground level. Wind shear is attributed with the change in wind speed, while wind veer is associated with the change in direction. Wind shear is often represented as the wind shear exponent α.

$$\alpha = \frac{\ln(v_2) - \ln(v_1)}{\ln(h_2) - \ln(h_1)}$$

(Eqn. 8)

where v_1 and v_2 are the wind speeds measured at the respective heights h_1 and h_2.

Wind veer is often represented by the equation,

$$\beta = \frac{\theta_2 - \theta_1}{h_2 - h_1}$$

(Eqn. 9)

where θ_1 and θ_2 are the wind directions measured at the respective heights h_1 and h_2.

Turbulence intensity is one of several parameters used to describe the atmospheric conditions at a given site, given as,

$$\text{TI\%} = \frac{\sigma_u}{\overline{\text{U}}} * 100\%$$

(Eqn. 10)

where σ_u is the standard deviation of the wind speed and $\overline{U}$ is the mean wind speed of the same sampling interval.

The bulk Richardson number (Ri_B) is a stability parameter used to describe the buoyant production of turbulence to the mechanical turbulence produced from the wind shear using temperature and wind speeds from two separate measurement levels. When wind speed and temperature are known at two measurement levels, the gradient Richardson number can be approximated as,

$$Ri_B = \frac{g(T_2 - T_1)(z_2 - z_1)}{\overline{\text{T}} * (u_2 - u_1)^2}$$

(Eqn. 11)

where the gravitational constant g = 9.81 m/s^2, T_2 - T_1 is the change in temperature at the different measurement levels z_2 and z_1, $\overline{T}$ is the mean temperature across the measurement levels and u_2-u_1 is difference in wind speed between the z_2 and z_1 (Zoumakis & Kelessis, 1991). The Monin-Obukov length (L), which describes the buoyancy effect on turbulence flows in the lower atmospheric boundary layer can be approximated using Ri_b. Given the direct correlation between the two, $\zeta = z/L = f(Ri_b)$, stability parameters can be established.

Providing site characterization using the parameters described above is an important step in wind energy development, both during early phase site-assessments but also during operation. As previously mentioned, the air temperature, pressure and relative humidity have a correlation to the air density. The air density has a direct correlation with the power produced by a wind turbine. Thus, it is to understand the role of atmospheric stability and air conditions to evaluate the impacts on the turbine performance.

2. Experimental Setup

This section provides details on UMD's innovative research facility for a small scale VAWT (Figure 2.1). One unique characteristic of the facility is that it is designed to measure the inner rotor fluid dynamics of a field scale VAWT, provide practical real-world data while also enabling in-depth exploration of fundamental fluid dynamic characteristics of complex vortex structures. Along with the inner rotor fluid dynamics, the facility enables investigating atmospheric conditions within the lower-level boundary layer that influence the turbine performance. These methods are completed by the strategic positioning of seven sonic anemometers, a pyranometer, rain gauge and a fine-wire thermocouple providing four measurement regions within the tower domain.

Figure 2.1:Aerial view of the UMD Research and Field Studies Center VAWT.

2.1. Vertical Axis Wind Turbine

The VAWT used in this setup is a 2.4 kW fixed straight blade H-Darrieus type, produced by Semtive marketed as the model Nemoi M version 1.0 (Table 2.1). It is mounted atop a 19.5m A53 Grade B steel tower located at the UMD Land Labs (GPS Coordinates; 46.871758, -92.057184) in Duluth, MN. A three-phase permanent magnet direct drive generator is mounted at the top of the tower. The blades are mounted directly to the generator using two aluminum alloy struts forming an "X" pattern (Figure 2.2). The blades have a unique, open profile low pressure side trailing edge that is likely to help with a low cut-in wind speed and provide load and rotational speed control at high wind speeds. When looking down on the rotor from the top, the blades rotate clockwise.

Table 2.1: Properties of the Semtive Nemoi M v1.0 VAWT and tower.

Rated Power	2.4 kW
Diameter	3 m
Hub Height	20 m
Blade Length	3 m
Swept Area	9 m^2
Cut-in Wind Speed	1.4 m/s
Rated Wind Speed	11.1 m/s
Maximum Wind Speed	58.1 m/s
Generator Type	Three-Phase Permanent Magnet 110/220 V
Blade Material	Aluminum Alloy
Tower Material	A53 Grade B Steel
Tower Height	19.5 m
Tower Diameter (base/top)	1 m / 0.63 m

Figure 2.2: Semtive Nemoi M VAWT v1.0 rotor model.

2.2. Data Acquisition

The DAQ system is one of the primary components to evaluate the fluid-structure interactions with the VAWT. This system takes high frequency wind speed and direction measurements at four different locations, including three different levels. Three 3D Gill WindMaster Pro sonic anemometers are located at 20 (6.1), 40 (12.2) and 60 (18.3) feet (meters) above ground level (AGL) which sample at 32Hz. A fourth 3D sonic anemometer (Campbell Scientific CSAT3BH) is located inside of the VAWT rotor at approximately one-half the radius of the turbine rotor, aligned with the bottom tip of the blade (approximately 2 inches, or 0.05m above the bottom tip), sampling at 50Hz (with maximum capabilities of 100Hz). Four additional sensors are included for atmospheric observations and redundancy. Three of these being two-dimensional anemometers (Campbell Scientific MetSENS500) which also record the barometric pressure, relative humidity, temperature, and magnetic heading located at the three respective levels sampling at 1Hz. A pyranometer (Campbell Scientific CS320) is used to measure the incoming solar radiation (385-2105 nm) at the mid-level (40ft) sampling at 0.2Hz. The user manuals of all the sensing equipment used in the development of the research facility can be found on Campbell Scientifics website.

2.3. Sensors and Dataloggers

The data acquisition system is designed to provide high-resolution measurements throughout the life of the turbine. It consists of two Campbell Scientific (CS) CR1000X data loggers capable of a wide range of digital and analog input channels with various communication protocols, sampling up to 1000Hz. Campbell Scientific has a proprietary coding language to program each datalogger. The specific code used at the facility is linked in Table 7.3Table 7.4Table 7.5 in the appendix. One datalogger is configured with all the CS sensors. A CSAT3BH heated 3D sonic anemometer with a fine-wire type E thermocouple located inside the rotor (Table 2.2)

The CSAT3BH and thermocouple provide high-resolution wind vector and temperature measurements, consistently sampling at 50Hz with an easy-to-implement 100 Hz sampling option. Three Campbell Scientific MetSENS500 sample at 1Hz providing the sonic wind speed and direction, temperature, relative humidity, barometric pressure, and magnetic heading with one at each of the three measurement levels (Table 2.3). A single Campbell Scientific CS320 digital thermopile pyranometer (Table 2.4) is located at the mid-level (40ft) of the tower measuring the net incoming shortwave radiation from 385-2105 nm at sampling 0.2 Hz. Completing the CS system is a 52202-L electrically heated rain gauge, which operates as a tipping bucket. It located atop the shed providing precipitation measurements at 0.1mm resolution (Table 2.5). The second datalogger is set up with the remaining Gill WindMaster 3D sonic anemometers, with one each at the three measurement levels. The Gill WindMaster Pro's model 1561-PK-020 are 3D sonic anemometers that sample continuously at 32 Hz providing a wind vector and sonic temperature (Table 2.6).

Table 2.2: Campbell Scientific CSAT3BH Heated 3D Sonic Anemometer Properties

Sampling Rate	1-100 Hz
Communication Mode	CPI
u_x, u_y, u_z	0-65 m/s
Ts (Sonic Temperature)	-40°C to 50°C
Thermocouple	Type E (0.003")

Table 2.3: Campbell Scientific MetSENS500 Compact Weather Sensor Properties

Sampling Rate	1 Hz (Max)
Communication Mode	RS-485 Modbus
Wind Direction	0 to 359°
Wind Speed	0.01 to 60 m/s
Temperature	-40°C to 70°C
Relative Humidity	0 to 100%
Barometric Pressure (Sea Level)	300 to 1100 mbar
Magnetic Heading	0 to 359°

Table 2.4: Campbell Scientific CS320 Digital Thermopile Pyranometer Properties

Sampling Rate	0.5 Hz (Max)
Communication Mode	SDI-12
Spectral Range (Net Shortwave Radiation)	385-2105 nm
Solar Radiation	0-2000 W/m^2
Sensor Temperature Operation Range	-50°C to 60°C

Table 2.5: Campbell Scientific 52202-L Electrically Heated Rain Gauge Properties

Measurement Rate	0.1mm/tip
Communication Mode	Pulse
Sensor Temperature Operation Range	-20°C to 50°C

Table 2.6: Gill WindMaster Pro 3D Sonic Anemometer Properties

Sampling Rate	32 Hz (Fixed)
Communication Mode	RS-485 Half-Duplex
u_x, u_y, u_z	0-65 m/s
Ts (Sonic Temperature)	-40°C to 70°C

2.4. Instrumentation Configuration

The sensors are strategically placed along the height of the turbine tower. This results with four distinct measurement zones on the tower (Figure 2.3). Starting at the top of the tower, as seen in Figure 2.4, inside the rotor the CSAT3BH is positioned about 0.5 m (20 in.) radially inward from the lower tip of the turbine blades. Extending outward from the tower 3.0 m (9.8 ft) in the same direction of the CSAT3BH is one of three WindMasters. Perpendicular from the CSAT3BH is one of three METSENS500 extending 0.8 m (32 in) from the tower. Located 6.1 m (20 ft) down from the top stations are the second of three WindMasters and METSENS500 and the CS320. The WindMaster is aligned with the upper one, but slightly inward extending 2.4 m (8 ft) from the tower. Midway along the same boom arm is the CS320 roughly 1.5 m (5 ft) from the tower and clear of any tower or rotor shadowing effects throughout the year. Likewise, the METSENS500 is aligned with the other and extends 0.74 m (29 in.) from the tower. At the lowest station on the tower, 6.1 m (20 ft) down from the mid-station, are the final WindMaster and METSENS500. The WindMaster is aligned with the two above it and extends 2.4 m (8 ft) from the tower. The METSENS500 is also aligned with the two above it and extends 0.74 m (29 in.) from the tower. Lastly, the 52202-L Rain gauge is mounted atop the shed approximately 6.1 m (20 ft) from the base of the turbine. The boom arms are 2" aluminum poles which are cross supported in all directions with guy wires as a redundancy. These were installed in an attempt to reduce the vibrations from the turbine and wind. All the mounting to the tower is through bolted to ensure the safety of the surrounding environment.

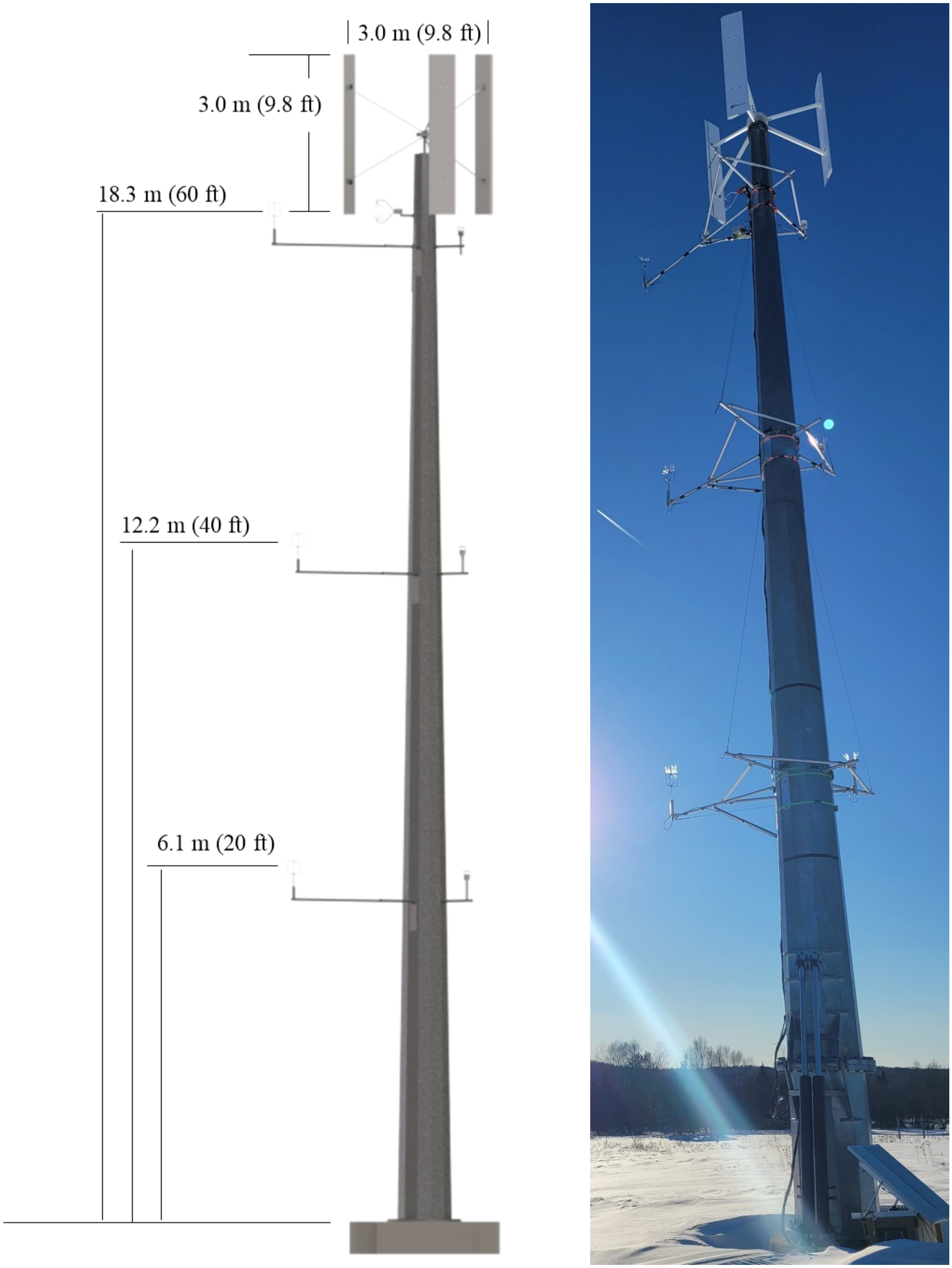


Figure 2.3: Tower and turbine rendering with dimensions (left) and field image from winter 2021/22 (right)

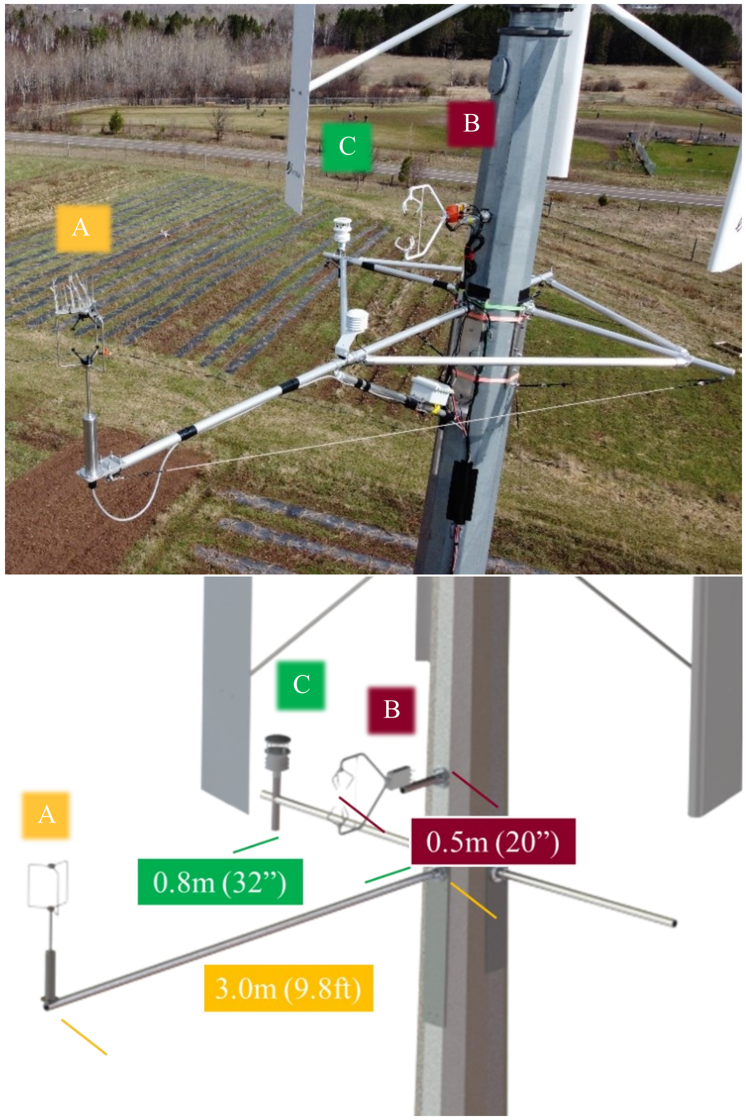


Figure 2.4: Top measurement level picture (top) and rendering (bottom) showcasing the WindMaster Pro (A), CSAT3BH (B), METSENS500 (C)

2.5. Orientation

The boom arms and each of the sensors were orientated in relation to the historic winds in the area (Figure 2.5). The Duluth International Airport (DLH) which is located approximately 13 km (8 miles) WSW from the UMD Wind Turbine (Figure 2.6 top) has historic data available dating back to 1970, providing hourly the wind speed and direction up to the current day. This data shows the prevailing winds are primarily from the West-Northwest (300°) or East-Southeast (120°) (Figure 2.6 bottom). Figure 2.7 shows the historic variability at DLH by month, which was anticipated to have a close correlation with the UMD Wind Turbine site. Based on the historic data the WindMasters were positioned normal to the South-Southwest (210°) face of the tower, providing quality data for much of the time. This positions the

CSAT3BH at a location to measure blade vortices during the blade upwind stroke when the winds approach from the West-Northwest (300°) and during the blade downwind stroke when the winds are from the East-Southeast (120°). Outside of the time when the CSAT3BH is positioned in the wake of the turbine tower, the unique locations provide an opportunity to measure inner-rotor blade vortex characteristics under a wide variety of atmospheric conditions and locations downwind from passing blades. The METSENS500 are positioned normal to the West-Northwest (300°) face of the tower, allowing for a reference when there is tower-shadowing on the WindMasters. Figure 2.5 provides an overhead view of the top sensor station and Figure 2.8 provides the compass orientation of the sensors. As directed by the manufacturers, all the WindMasters, METSENS500 and CS320 are aligned to approximately North with their respective on-sensor markers, while the CSAT3BH is co-linear with its boom-arm. Further detail on the sensor alignment can be found in section 3.1.

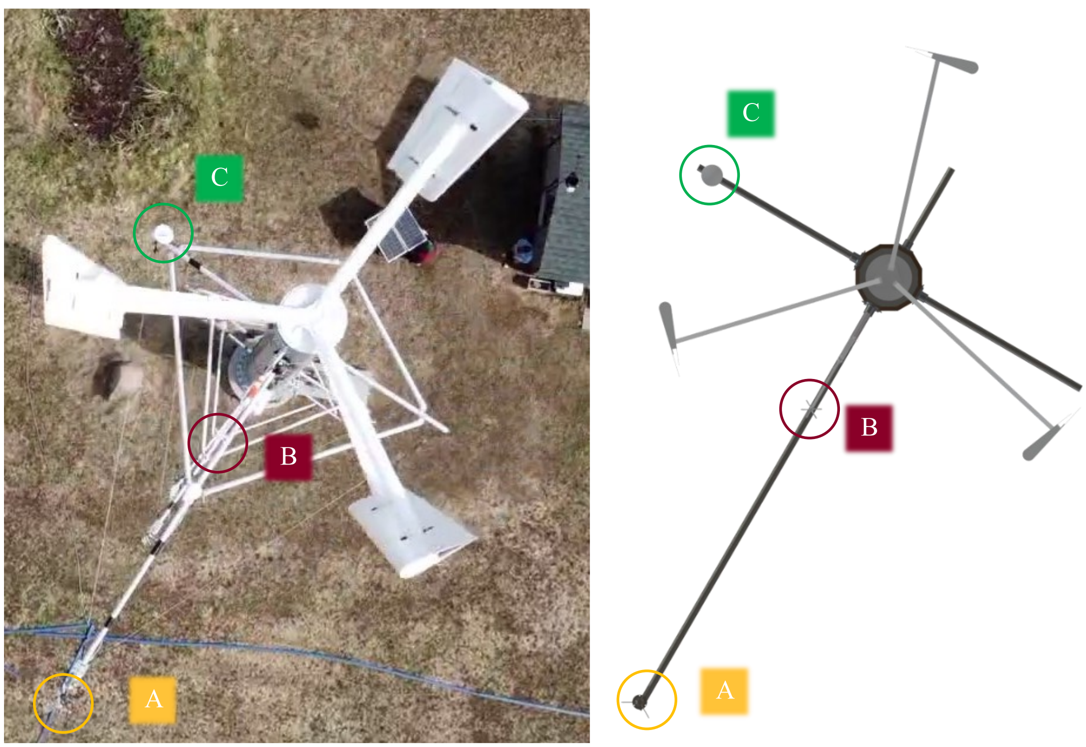


Figure 2.5: Top view of the turbine (left) and rendering (right) showing the WindMaster (A), CSAT3BH (B), and METSENS500 (C).

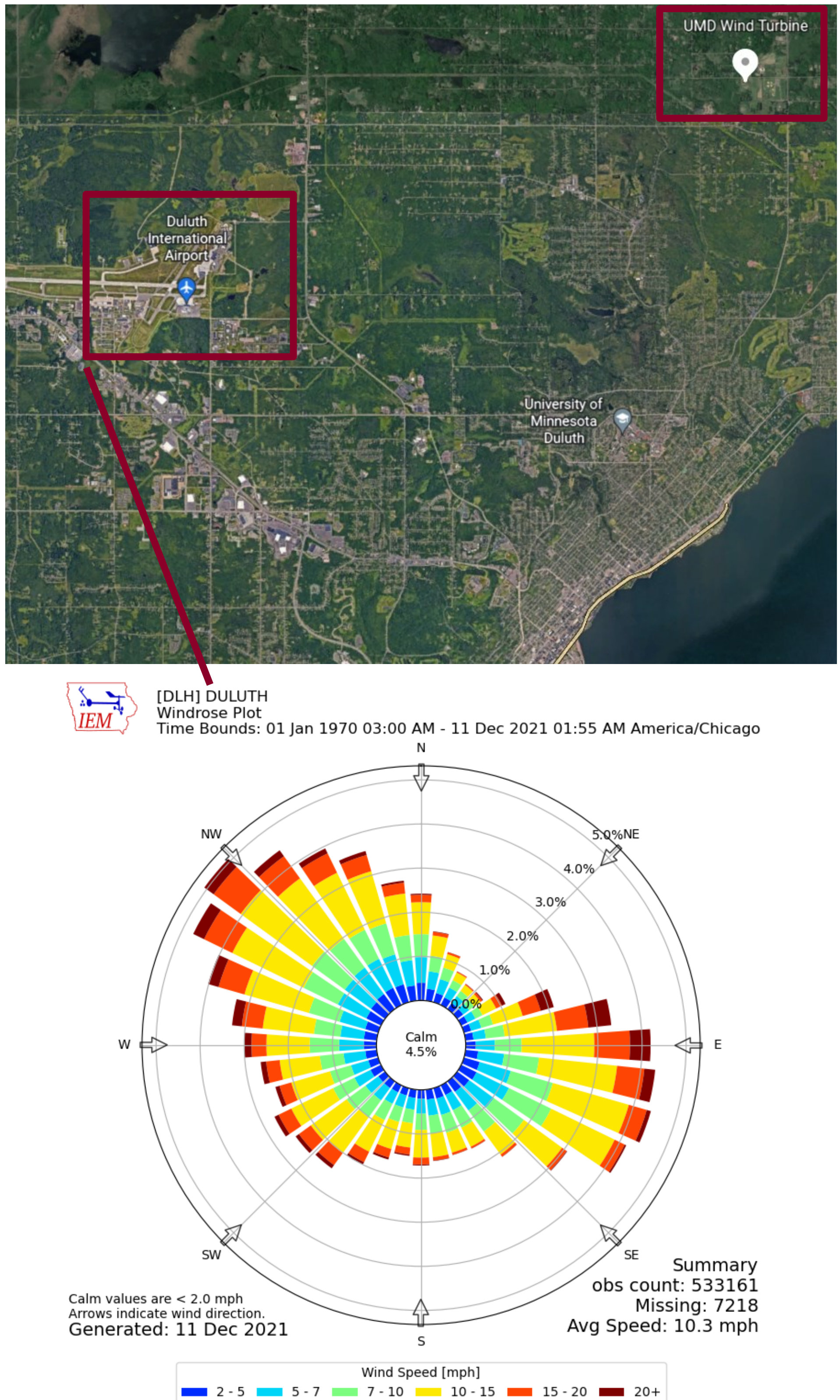


Figure 2.6: Historic winds at DLH (bottom), primarily from the NW or E-SE, with a map (top) for reference to the UMD Wind Turbine with data from January 1970 to December 2021.

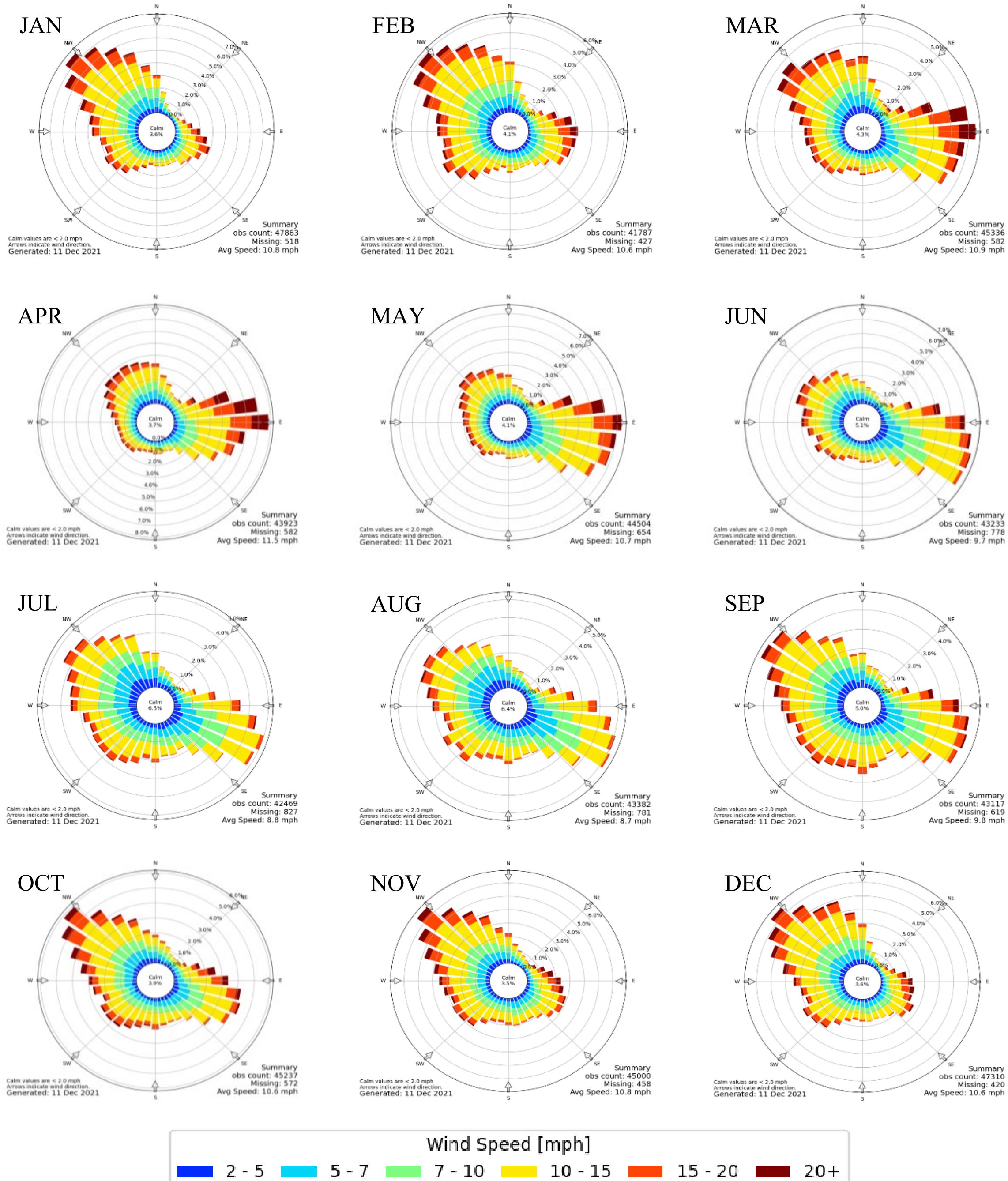


Figure 2.7: Historical wind roses for DLH by month with data from January 1970 to December 2021.

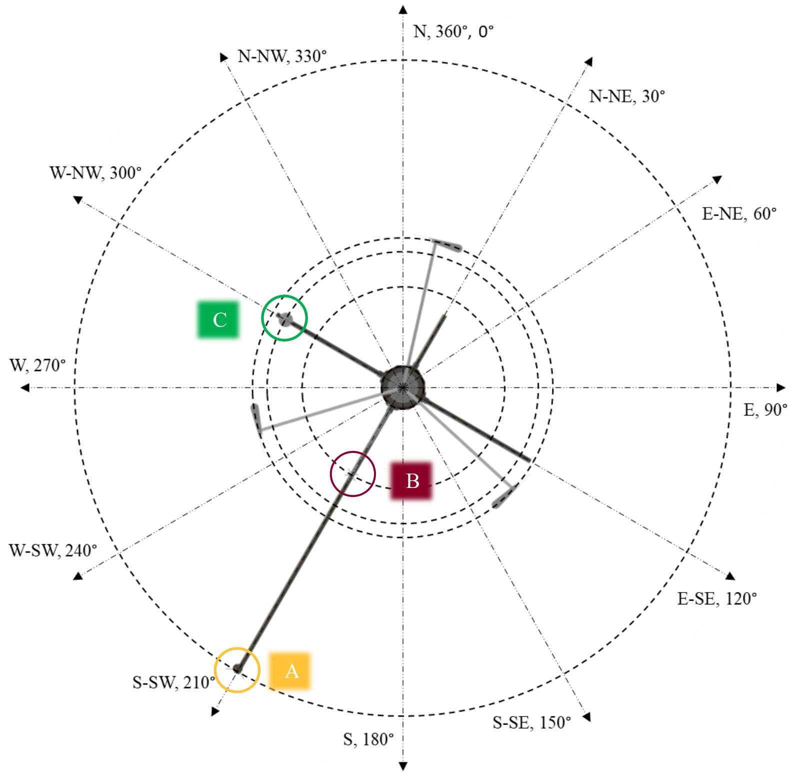


Figure 2.8: Compass orientation of the sensors in correspondence with the wind turbine WindMaster (A), CSAT3BH (B), and METSENS500 (C).

3. Results

The results presented in this section illustrate and quantify the unique capabilities the state-of-the-art VAWT facility at the UMD Research and Field Studies Center can provide. The data acquisition system runs continuously and provides reliable data sets when not interrupted. Throughout the development of the system, some data was lost while working to optimize the reliability and collection. It should be noted that when the CSAT3BH is scanning at 100Hz, the MetSens500 data is unavailable. Therefore, there are only a select few data sets with the high-resolution 100Hz data (regularly scans at 50Hz).

3.1. Alignment

Corrections to the wind direction and speed are applied at each station to account for misalignment that occurred during installation. The MetSens500 wind direction is corrected with the magnetic heading which is recorded by each sensor. For each record the correction is applied with the averaging being -21°, -6.5° and 9° from North at the respective bottom, middle and top stations (Table 3.1). The WindMaster corrections of 5°, -5° and -5° are applied based on the MetSens500 corrections at the respective locations. The corrections are applied with a data set from March 3, 2022 over the 15:00 hour (Figure 3.1 and Figure 3.2) which the wind was primarily coming from the South South-West (SSW) to West and increased from 5 m/s to 8 m/s.

Table 3.1: MetSens500 data corrections based on 3600 samples and tower shadowing approximations.

MetSens500	A – Bottom (deg.)	B – Middle (deg.)	C – Top (deg.)
Mag Heading Range	336-341	352-356	8-12
Average Mag. Heading	338.968	353.975	9.996
Average Degrees from N	339 (-21)	354 (-6)	10
Tower Shadowing	~105-145	~115-135	~120-130

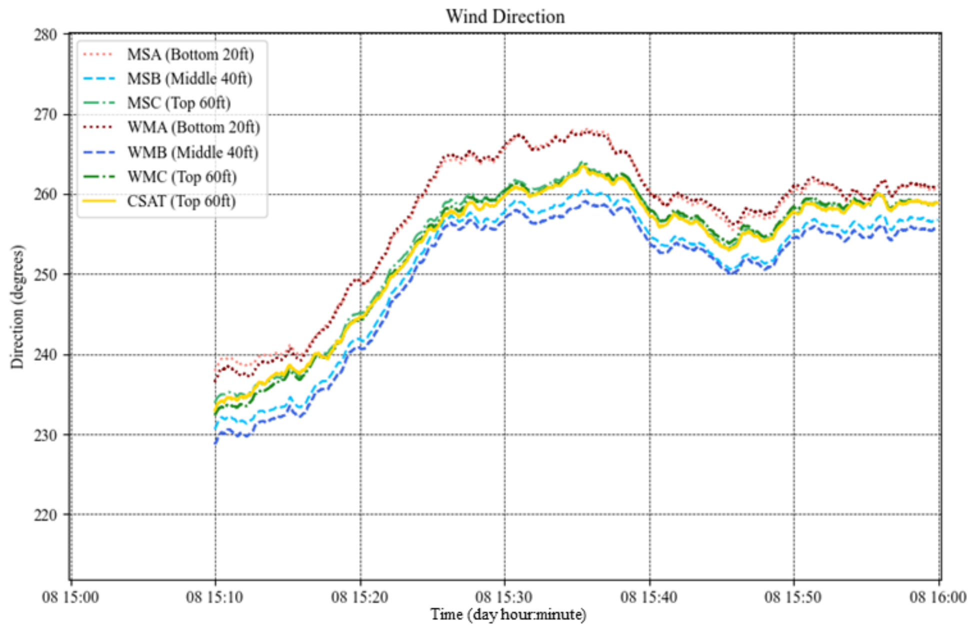


Figure 3.1: 10-minute rolling wind direction average over the 15:00 hour on March 8, 2022, highlighting the agreement between the various sensors and stations.

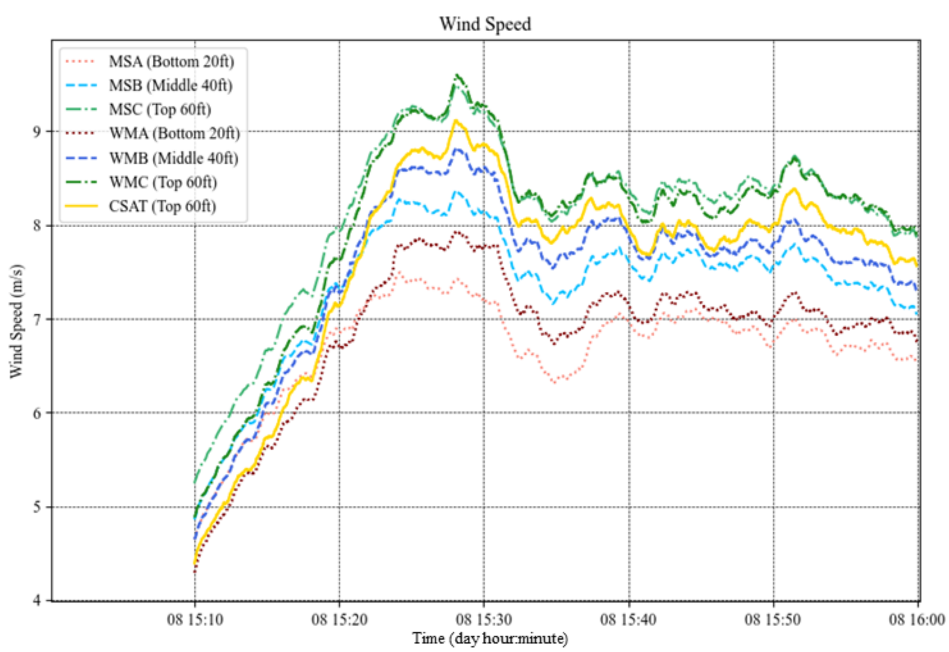


Figure 3.2: 10-minute rolling horizontal wind speed average over the 15:00 hour on March 8, 2022, highlighting the agreement between the various sensors and stations.

The Pearson Correlation Coefficient (PCC) was found to verify a strong relationship between the MetSens500 and WindMaster 3D sonic anemometers in the wind direction and speed. A PCC r-value above 0.9 implies there is a very strong correlation in the data. The resulting PCC r-values of 0.982 (bottom), 0.994 (middle) and 0.988 (top) were found for the wind direction. Similarly, the PCC r-values for wind speed of 0.965 (bottom), 0.981 (middle) and 0.983 (top) were achieved (Table 3.2, Figure 3.3, Figure 3.4). The correlation between the MetSens500 and the CSAT was also found in the wind direction and speed at r = 0.994 and r = 0.981, respectively. Using other data which the prevailing winds were primarily from the East to South, the approximate tower shadow regions were found when the PCC r-value fell well below 0.9. The three regions of tower shadow vary based on the tower diameter, being tapered from the bottom to top (Figure 3.5).

Table 3.2: Pearson Correlation Coefficients for sensor corrections on March 8, 2022, with 61 samples averaged at 1-minute intervals.

Sensor Station	A (Bottom)	B (Middle)	C (Top)	CSAT (Top)
Pearson Correlation Coefficient Wind Direction	0.982	0.994	0.988	0.994
Pearson Correlation Coefficient Wind Speed	0.965	0.981	0.983	0.981

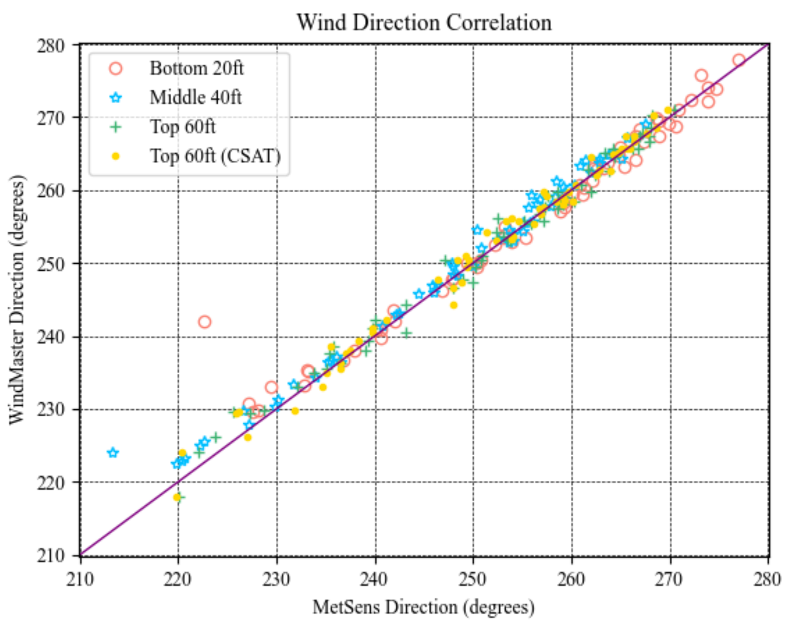


Figure 3.3: Wind direction correlation of one-minute average intervals for the 15:00 hour of Match 8, 2022 with wind speed from 3-13 m/s.

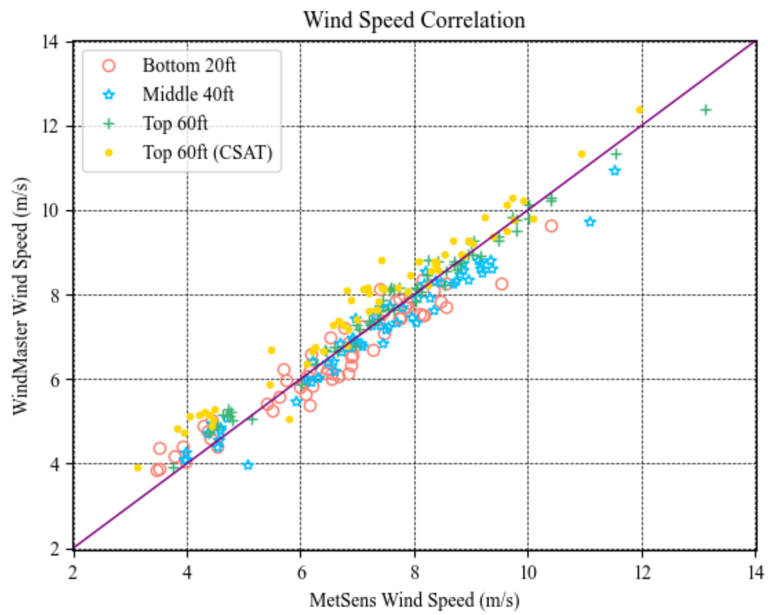


Figure 3.4: Wind speed averaged at one-minute intervals for the 15:00 hour of Match 8, 2022 with the winds primarily from the W-SW.

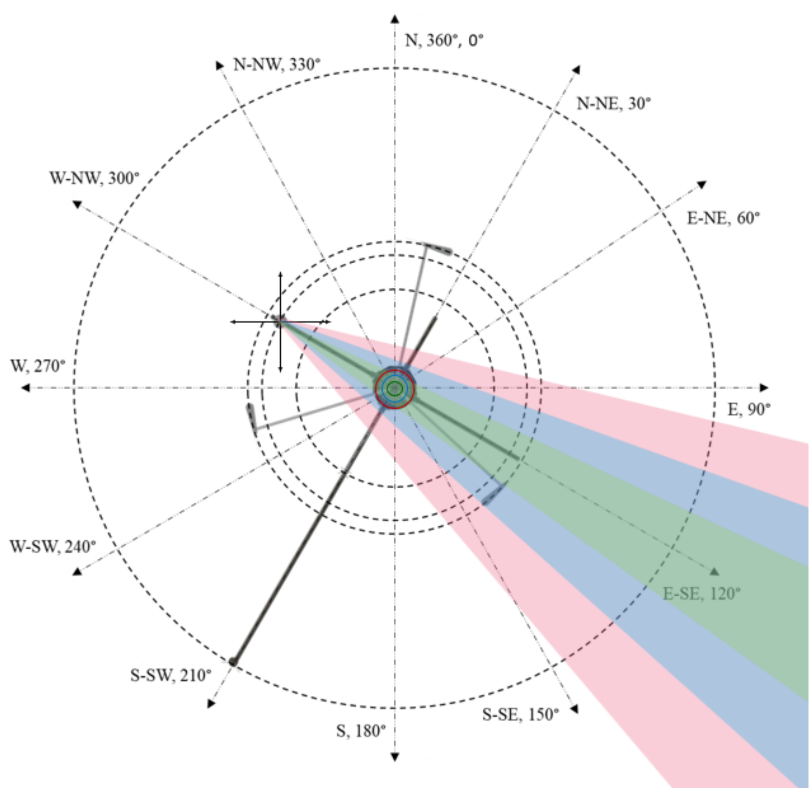


Figure 3.5: Approximate tower shadowing regions on the MetSens500s (MS) at each level. The red (largest) region is the zone in which the bottom MS compact weather station anemometer will record tower-shadow contaminated wind speed and direction data if the winds come from a direction within the red-shaded zone. The blue (medium) corresponds to the middle MS (12.2m), and the green (smallest) corresponds to the top MS (18.3m).

3.2. Baseline Meteorological Conditions

A variety of atmospheric observations in Figure 3.6 shows the capabilities of the tower setup, with the ability to classify the changes of meteorological conditions throughout a given period. The selected data is illustrated with a 5-minute moving average, being selected to provide a clear image of how the various atmospheric conditions vary throughout the day. This period of 24-hours over March 8, 2022, shows a warm front (low pressure) moving into the region with a near 10°C increase and temperature, 40% decrease in relative humidity and a shift in the wind direction from South-Southwest to West-Northwest.

The short-wave radiation is captured over the same period, seen in Figure 3.6, with the CS320 pyranometer. The solar irradiance has a clear relationship with the air temperature over the same period as seen in. The turbulence intensity is calculated over the same period (Figure 3.7) to help characterize the wind conditions. The TI is fairly consistent throughout the day with more variation beginning at 15:00hr after the warm front arrived causing a drop in humidity and shift in wind direction and speed.

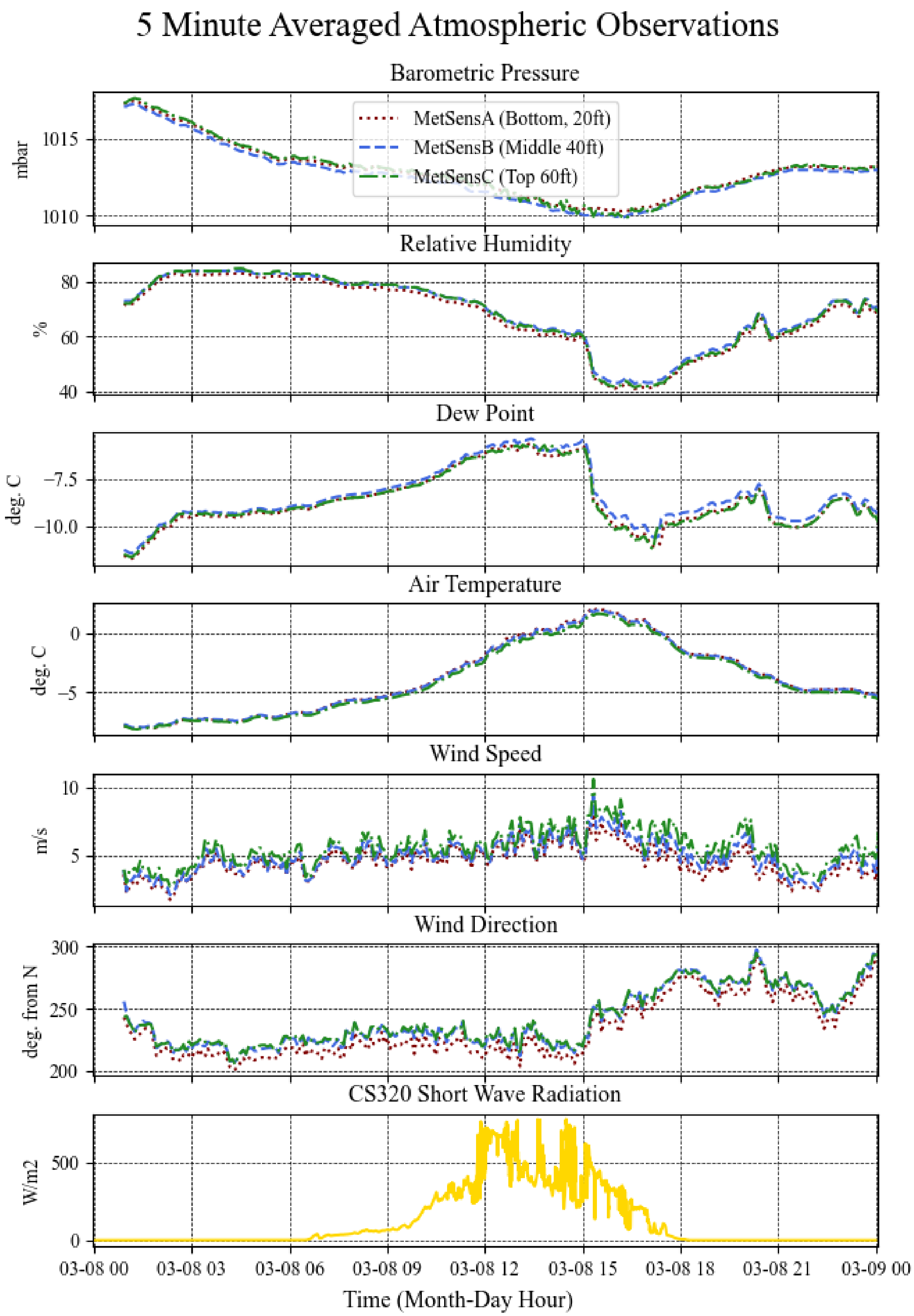


Figure 3.6: Atmospheric observations throughout March 8, 2022, with a 5-minute rolling average sampled at 1 Hz, capturing the extreme variability in weather conditions at the site.

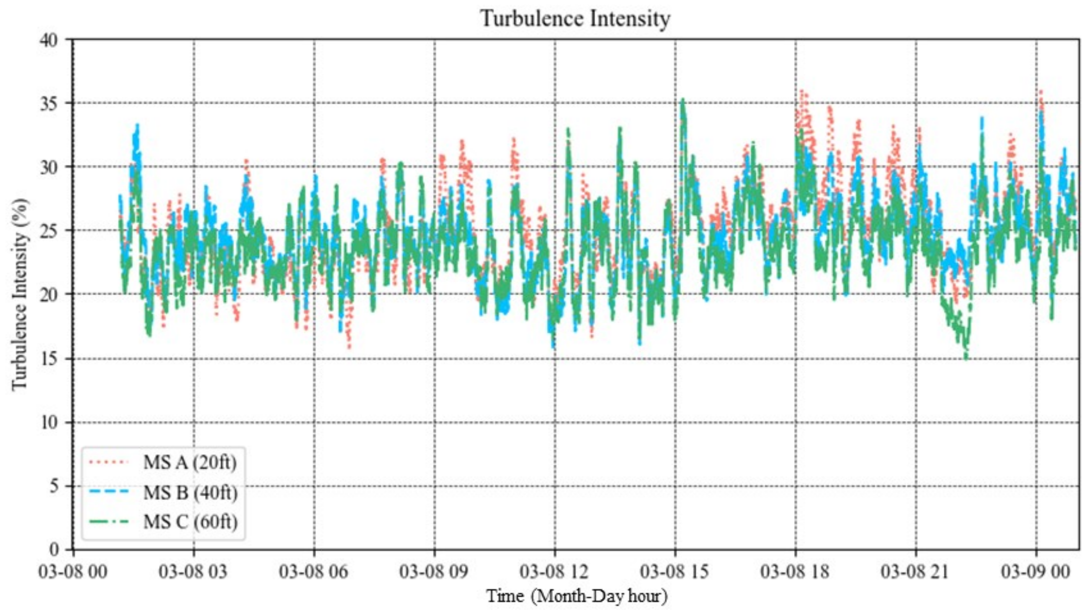


Figure 3.7: Turbulence intensity at each MetSens500 calculated from a 10-minute rolling average window, sampled at 1Hz throughout the day of March 8, 2022.

This period is further investigated during the 15:00hr of this period with four different averaging intervals, evaluating the wind speed, wind direction and turbulence intensity. The four averaging intervals are 1-minute (Figure 3.8), 2-minutes (Figure 3.9), 5-minutes (Figure 3.10) and 10-minutes (Figure 3.11) in which the data was resampled at its respective interval. The various intervals provide insight on how rapidly the wind shear and veer phenomenon occur and the major fluctuations in the turbulence intensity at the site. The 1 and 2-minute intervals illustrate wind veer of approximately 3.5 m/s while the 5 and 10-minute intervals illustrate wind veer of approximately 2 m/s from the lowest to highest stations. Similarly, the wind shear can be seen to fluctuate rapidly, but is more consistent throughout the averaging intervals with an approximate 5-degree shift. All four intervals show a consistent shear from the lowest station to the middle and upper stations which are relatively equal. Lastly, the 1-minute averaging window shows turbulence intensity fluctuations from about 10-40% where the 10-minute fluctuates from roughly 22-32%.

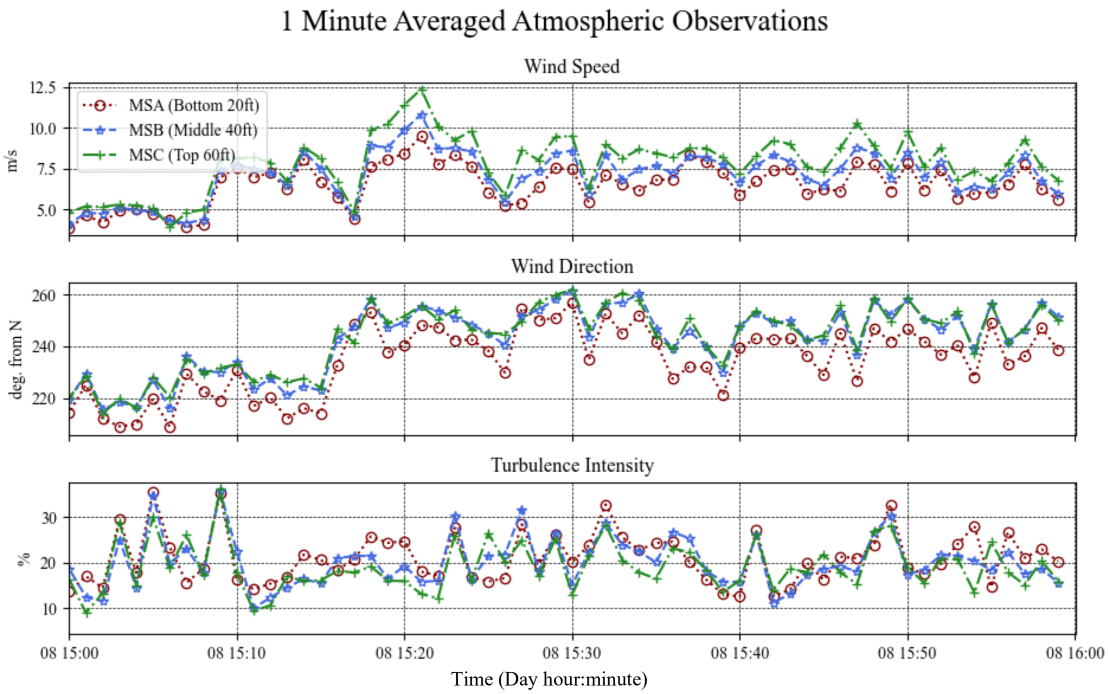


Figure 3.8: Atmospheric observations resampled to 1-minute averages over the 15:00hr of March 8, 2022, by the MetSens500s sampling at 1Hz at each station.

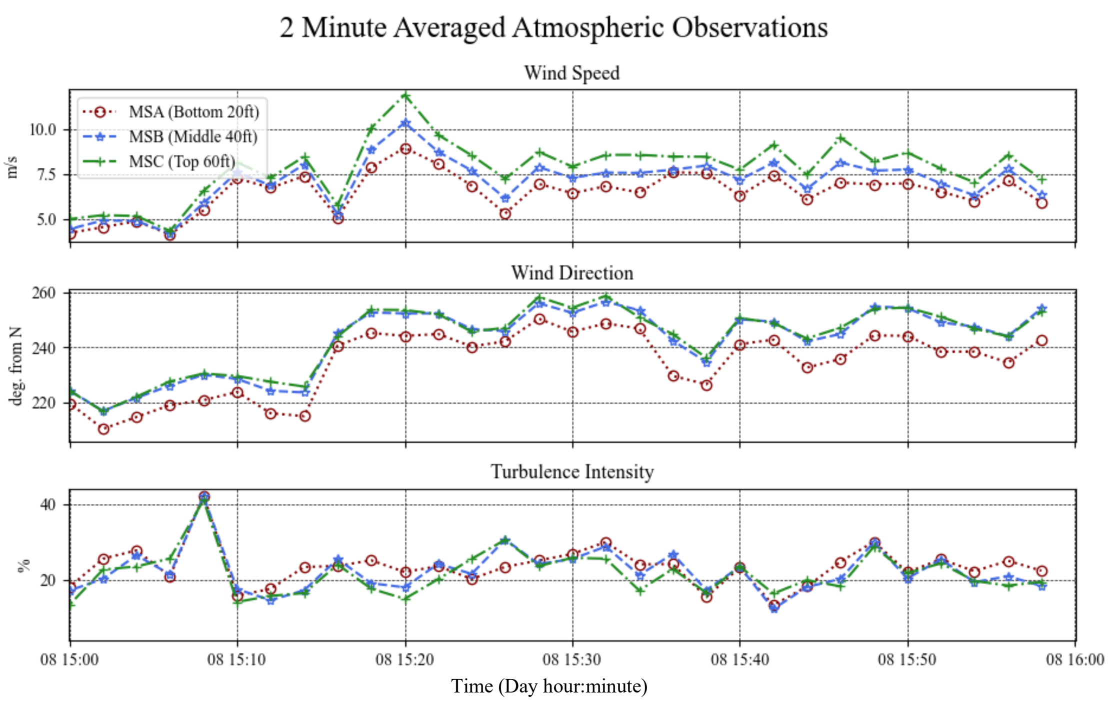


Figure 3.9: Atmospheric observations resampled to 2-minute averages over the 15:00hr of March 8, 2022, by the MetSens500s sampling at 1Hz at each station.

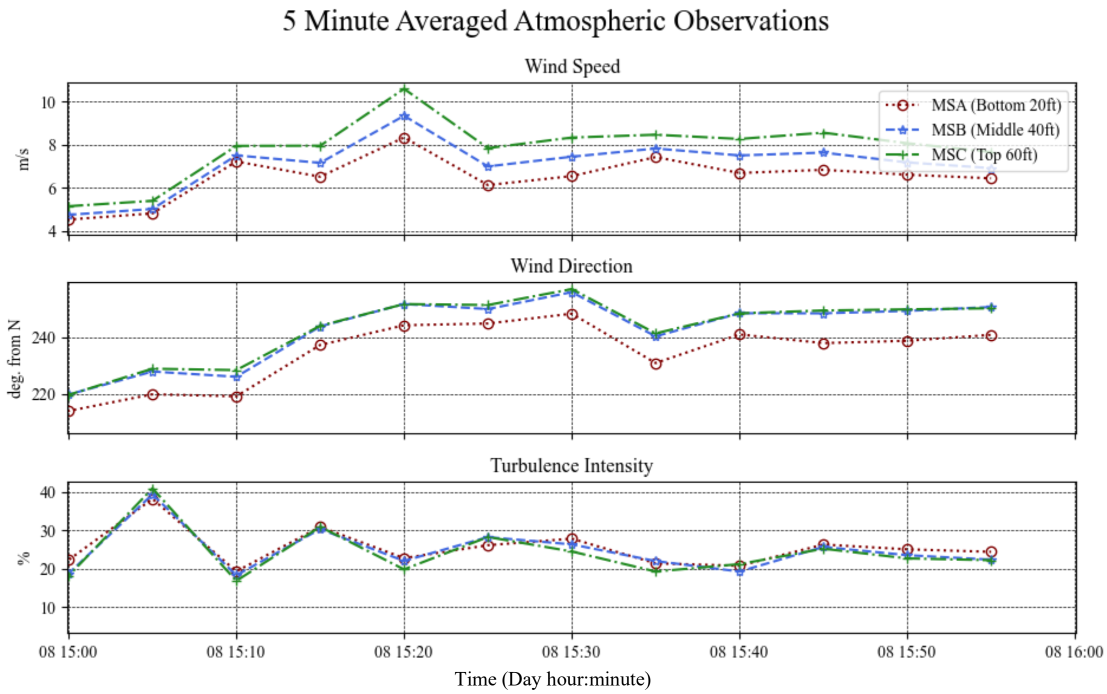


Figure 3.10:Atmospheric observations resampled to 5-minute averages over the 15:00hr of March 8, 2022, by the MetSens500s sampling at 1Hz at each station.

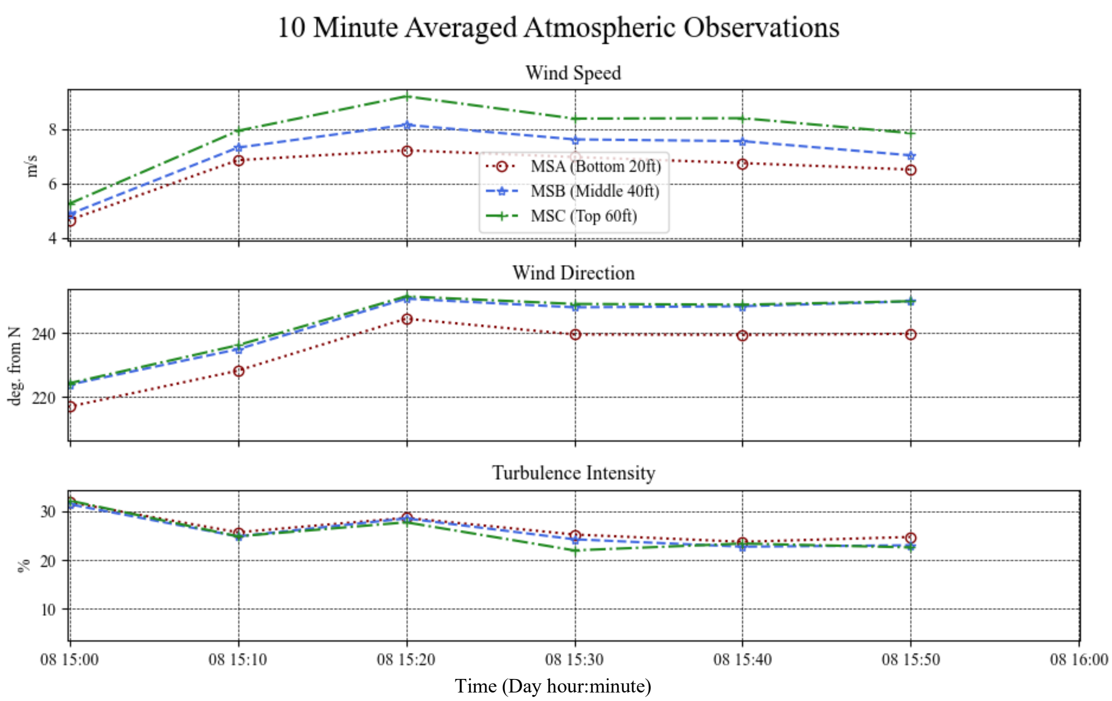


Figure 3.11:Atmospheric observations resampled to 10-minute averages over the 15:00hr of March 8, 2022, by the MetSens500s sampling at 1Hz at each station.

3.3. Inner Rotor Observations

Inner rotor observations were taken for a limited time with the turbine running. A period of 10 seconds on May 20, 2022, from 15:15:15 to 15:15:25 is examined (Figure 3.12), with an average windspeed of 10.7 m/s from the West-Southwest. The observations are made by the CSAT sampling at 100Hz, with the WindMaster as the reference flow sampling at 32Hz. Looking at the 2-second window, there is a clear disturbance in the wind every 0.33 seconds or at approximately 3Hz (Figure 3.13). This disturbance shows the flow separation of vortices from each blade passing the CSAT. It is estimated that the turbine is operating at about 60rpm during this period, resulting in an approximate $Re=6.4\cdot10^5$ and TSR=0.29. The blade revolutions were counted on a high-resolution video with similar wind conditions. Plotting the horizontal and vertical winds speeds against each other show a circular behavior (Figure 3.14), further indicating the vortex shedding from the turbine blade on each pass by the CSAT.

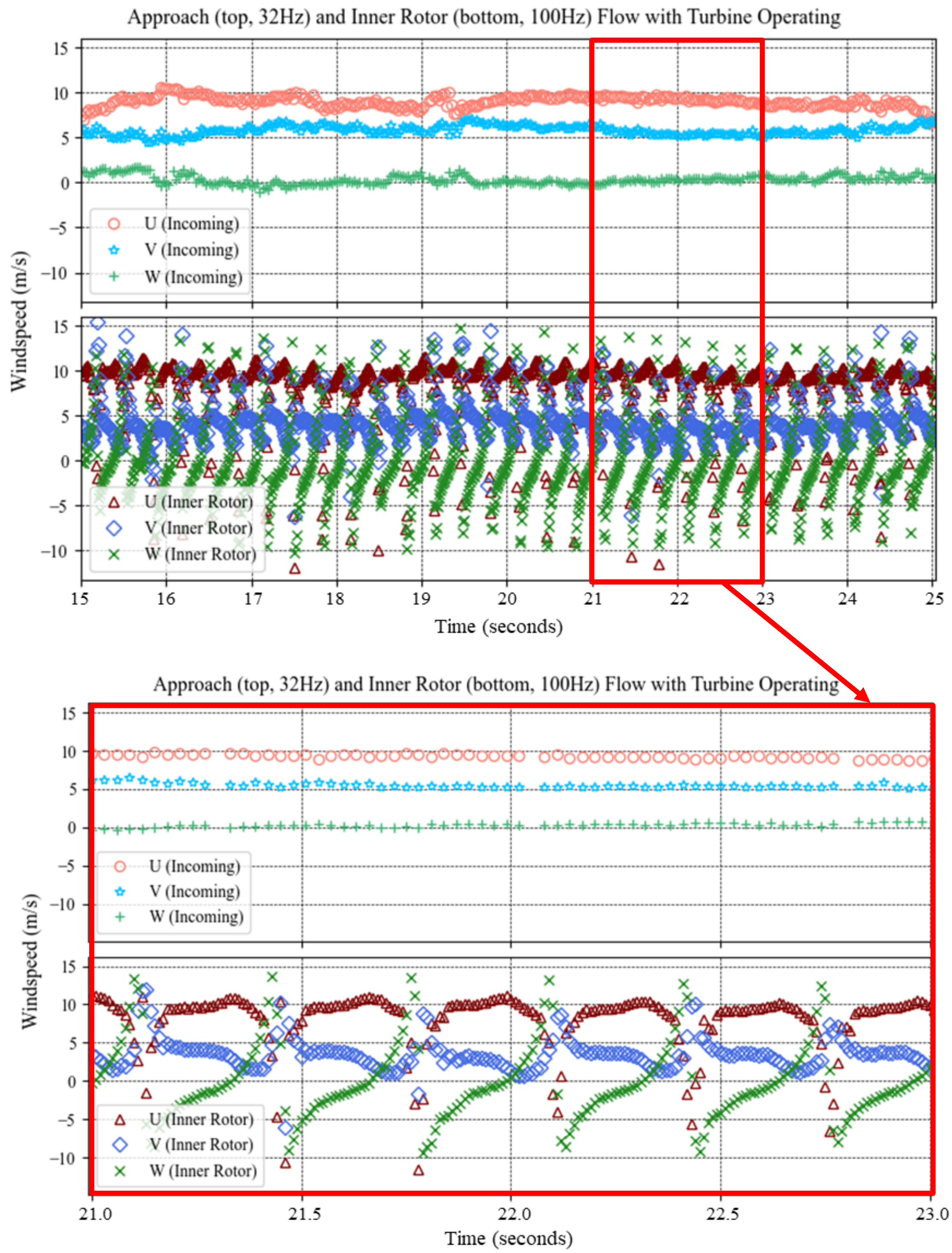


Figure 3.12: Ten second timeseries (top) at the top station showing the incoming flow to the WindMaster (32Hz) and the inner rotor flow of the CSAT (100Hz) with the turbine operating with a close-up examination of two seconds (bottom), where a spike in the data signifies a blade passing by the CSAT.

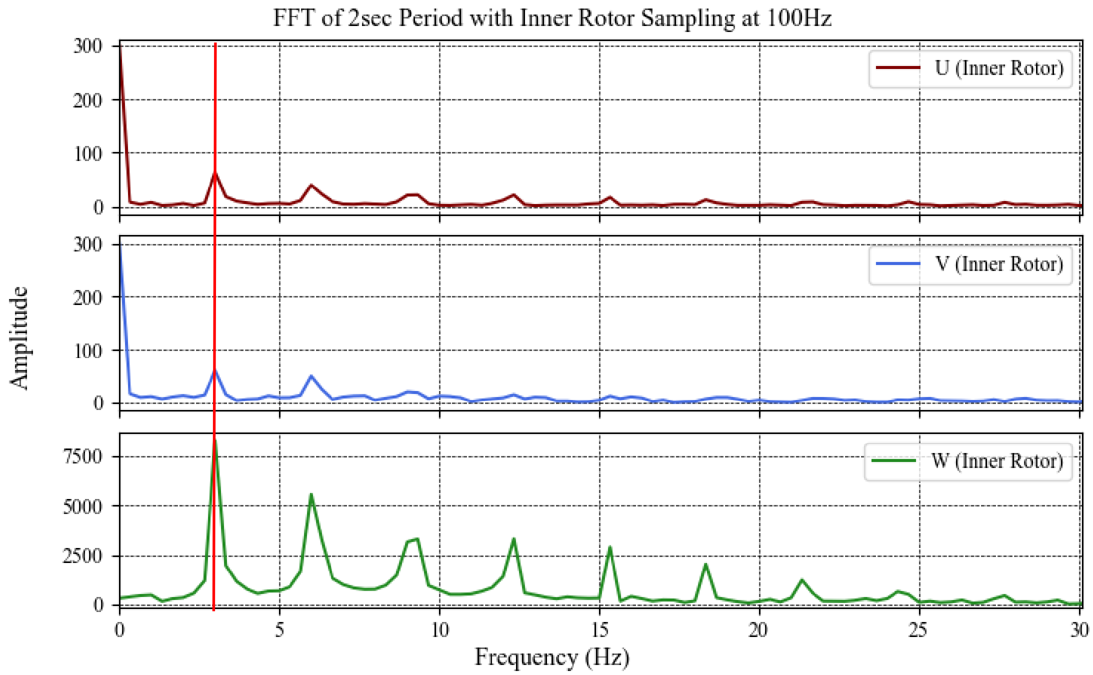


Figure 3.13: Inner rotor observations by the CSAT at 100Hz with the turbine operating, capturing the frequency of the vortices from the turbine blades, with the vertical red line at 3.33Hz representing the passing of each blade.

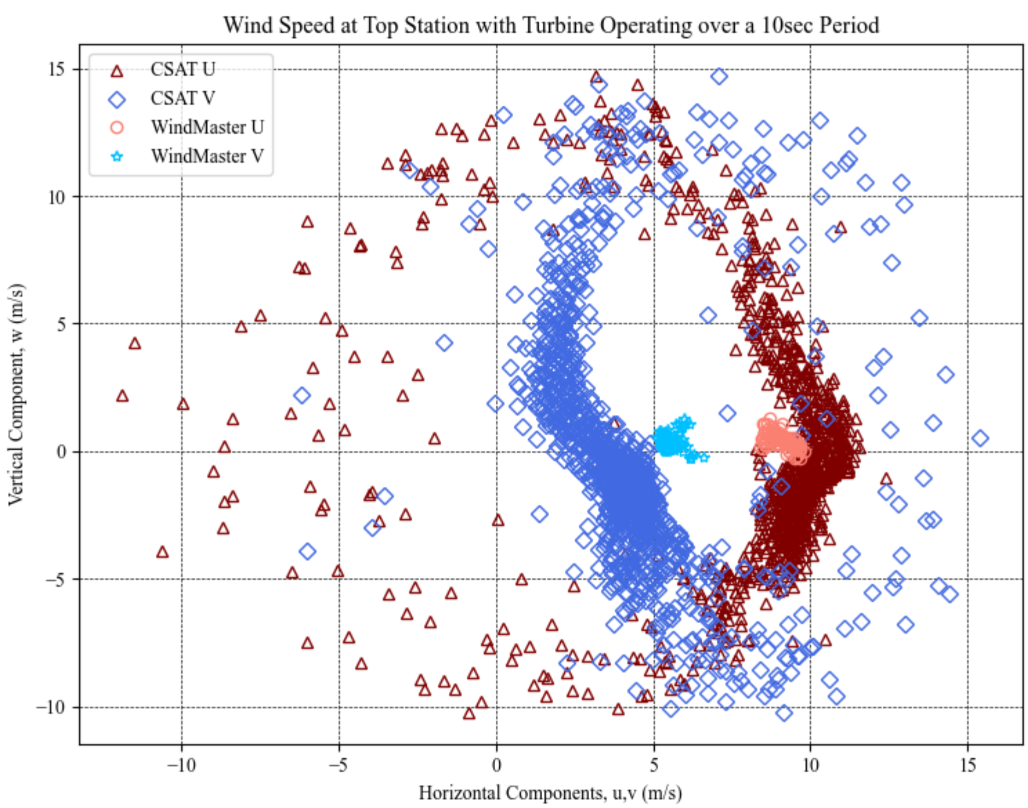


Figure 3.14: Representation of the circular flow characteristics measured by the CSAT (100Hz) inside the turbine rotor, compared to the inflow (32Hz) for a 10 second period.

4. Discussion

The VAWT experimental setup at the University of Minnesota Duluth has been developed to commence numerous VAWT research studies. It's unique and capable data acquisition system allows for high-resolution observations in the lower-level atmospheric boundary layer with room for additional sensor to be added. With this setup, the facility will be able to influence the VAWT sector and contribute to the growth of renewable energy. Energy demand is expected to grow roughly 80% by 2050 and renewable energy, specifically wind energy, is projected to play a critical role in the world's energy generation mix. It is projected that wind energy contributes up to 35% of the generation mix by 2050 (Energy Agency, 2021).

While much of the commercial market share is taken by large scale HAWTs, there is a renewed interest in VAWTs for distributed wind and offshore applications. Characteristics such as omni-directionality, performance increases with array clustering, ease of manufacturing and a base level located generator (specifically for floating offshore wind) are helping to drive this renewed interest. While there are still design challenges with VAWTs, such as the variable lift-to-drag ratio in the Reynolds transition region and lower Cp (when compared to a traditional commercial HAWT), among others outlined by Alessandro Bianchini et al.(Bianchini et al., 2022), the facility can influence some of these design challenges with small and medium scale VAWTs. The field scale setup allows for Reynolds number scalability from CFD and wind tunnel studies with the potential to extend to large scale VAWT applications.

Given the unique setup of sensors, the facility can capture an atmospheric profile in the lower-level atmospheric boundary layer with measurement stations at 6.1m (20ft), 12.2m (40ft) and 18.3m (60ft). The dataloggers allow for an abundance of inputs, providing the ability to add sensors for new or improved measurement capabilities. This also provides opportunities to be a cross-functional facility with other research disciplines, such as hydrology, agriculture, acoustics, among many others.

The preliminary investigation discussed in this paper aims to provide insight into some of the potential of the research facility. An analysis on the sensor alignment has been completed for the quality assurance and quality control of the data collected, resulting in a Pearson Correlation Coefficient r-value greater than 0.9. A look into the atmospheric conditions at the site has taken place to illustrate how the field-scale setup is unique from others and is distinguished from wind tunnel experiments. With the location of the wind turbine in Duluth, MN the facility can experience drastically changing weather patterns throughout a day. One example of this is presented, highlighting a low-pressure warm front moving through the area causing an extreme shift in the atmospheric observations. This provides the ability for the research impact a wide range of climates. Finally, a look into the inner-rotor fluid dynamics was conducted to guide the interests in fluid-structure interactions of VAWTs and the complex dynamic stall phenomenon which occurs. By sampling at 100Hz with a West-Southwest wind, vortex like observations were captured inside the rotor while the turbine was operating at approximately 60rpm.

4.1. Sensor Alignment

The sensor alignment is critical for a successful deployment of the UMD Research and Field Studies Center wind turbine research facility. When mounting the instruments, the turbine was in the lowered position, as seen in Figure 1.2, which was more than 90° from the upright. The tower also tapers from bottom to top with the boom arms mounted flush against the face of the turbine sides, meaning they are at a slight angle up from the horizontal when the tower is in the upright position. Along with the arms being mounted flush to the face, they're also on the 210° and 300° faces from North. Given these complications, the instruments were best aligned to North with the tower lowered providing a large sense of uncertainty.

Due to the error and uncertainty in the sensor alignment with corrections of -21° (MSA), -6° (MSB) and 10° (MSC) based on the magnetic heading, extra precautions should be taken to get all the sensors aligned. While the Metsens500s have a magnetic heading indicator, it may not be reliable to provide a good estimate of the misalignment due to the proximity to the steel tower which can affect the internal electric compass. The WindMasters were aligned in the same manner as the MetSens500s with corrections of 5°, -5°, -5° applied based of the respective MetSens500 corrected sensor data. The data corrections are deemed reliable based off a Pearson Correlation Coefficient greater than 0.9 which was found when applying the data corrections.

It's recommended that the three Metsens500s are moved to the N-NE (30° from North) boom arm of the tower. This boom arm was not originally installed when the Metsens500 locations were determined. With the prevailing winds being from the Northwest and East, this will allow for a better evaluation of the wake conditions with the WindMasters. The WindMasters are located approximately one rotor diameter from the turbine, with prevailing winds from the Northeast an investigation to the wake dynamics can be conducted. All the sensors need to be checked for levelness and properly aligned to North while the tower is in the upright position. There is a discrepancy in wind speeds and directions (Figure 3.1 and Figure 3.2) between the Metsens500s and WindMasters; an unlevel sensor will provide inaccurate data on the windspeed and direction. Some of these differences may be attributed to vibration and movement in the boom arms.

4.2. Atmospheric Observations

As mentioned earlier, there is a lack of field scale VAWT research studies completed, but a fair amount of wind tunnel experiments. This setup allows for in-depth analysis of how real-world atmospheric conditions influence a VAWT, and more specifically, in the small (residential) and medium (community) scales but with the possibility to apply to large (commercial) scale VAWTs. The preliminary investigation provides some insight into these capabilities with a look into the daily conditions experienced at the site and a few parameters that influence turbine performance.

The location of the UMD wind turbine at the Research and Field Studies Center experiences a wide range of atmospheric conditions. The historical maximum and minimum temperatures at the site record are 41.1°C (106°F) and -40.6°C (-41°F), respectively with changes being more than 17°C (30°F) in a day.

Along with extreme temperatures, the site faces icing events including rain, sleet and snow, and it is not uncommon to see the conditions change drastically throughout a day. At times the atmospheric conditions can be like an arctic or tropical region and everywhere in between. An example of the variability is provided in Figure 3.6, when a warm front moves through the region resulting in a major shift (roughly 50°) in wind direction. The rapid shifting of wind direction helps to show how important the omnidirectionality of a VAWT is in such regions.

While the TI was investigated at the site over several periods, a look into the Bulk Richardson number and Monin-Obukhov length did not provide any notable results. This should not be forgone as atmospheric stability has a correlation with the energy output of a wind turbine (St Martin et al., 2016). When evaluating the wind shear, veer, Bulk Richardson number, Monin-Obukhov Length and turbulence intensity different stability conditions can be found. St. Martin, et al. conducted a study on a traditional General Electric 1.5MW HAWT which farther outlines the over-performance of the power curve with different stability conditions (St Martin et al., 2016).

By examining different averaging periods in Figure 3.8 through Figure 3.11, it can be seen how estimates pertaining to the turbine's operation can be skewed. This small investigation to the averaging windows shows how rapidly the TI can change and that averaging at a long interval may dismiss the extent of the actual conditions the turbine is experiencing. By switching from 1-minute to 10-minute averaging, the TI is reduced from an approximate 10-40% window to roughly 22-32% window, where the latter would imply there are much calmer conditions. Goit et al. conducted a full study on the averaging windows suggesting 30 or 60 second windows for turbines with rotor diameters less than 15m. The power loss estimation from the 30-second to 10-minute window in the study showed an 8% underestimate (Goit et al., 2022).

Möllerström et. al illustrate a similar scenario on the effect of the averaging window and the influence the turbulence intensity in Figure 4.1. This figure shows how a smaller averaging window (1-minute) results in a higher average power coefficient, when compared to the standard (10-minute) averaging window on a 200kW turbine (derated to 80kW) (Möllerström et al., 2016). Based on the 1, 2, 5 and 10-minute averaging windows previously presented (Figure 3.8-Figure 3.11) similar results would be expected from UMD's 2.4kW turbine. Furthering the knowledge gap is the disparity between field scale and wind tunnel research, given wind tunnel studies are seemingly fall under 10% TI (Balduzzi et al., 2020; Brownstein et al., 2019; Li et al., 2016).

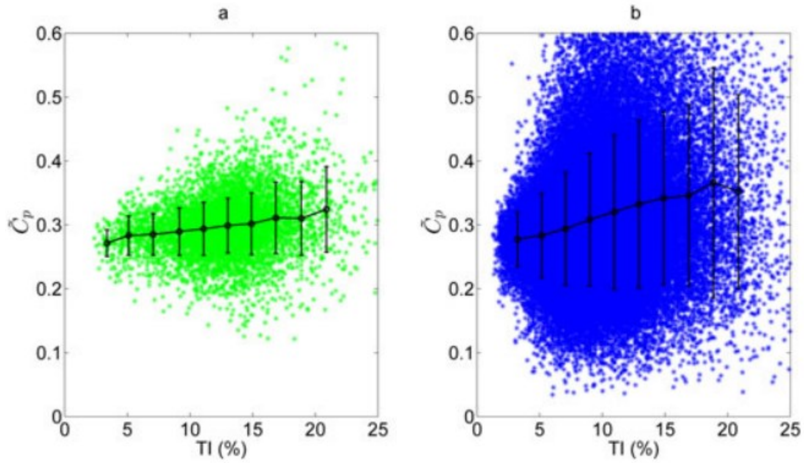


Figure 4.1: Turbulence intensity (TI) influence on C_p from 10 min mean values (Figure 6(a)) and 1 min mean values (Figure 6(b)) from (Möllerström et al., 2016) highlighting the dilution from averaging.

4.3. Inner Rotor Observations

A preliminary investigation to the inner rotor fluid-structure interactions has been conducted, given that vortices are present with each pass of a blade the CSAT3BH was able to sample data representative of these. The vortices can be visualized every 3Hz during a period with an average wind speed of 10.7 m/s from the West-Southwest with an approximate Re=$6.4 \cdot 10^5$ and TSR=0.29. This is a result of the rapidly changing angle of attack in relation to a relatively stable freestream wind velocity (Figure 3.12 bottom). As previously shown (Figure 1.14, Figure 1.15), while a blade is rotating it is producing lift and the AoA is changing. While the AoA is increasing there is a separation between the fluid-structure which is the beginning of a vortex. As the AoA continues to increase the vortex grows to the point in which the blade is no longer producing lift, initiating the dynamic stall phenomenon. With the AoA continuing to increase, the vortex begins to shed from the blade until it has completely detached. Without a further investigation into the vortices, it is unclear of the stage and size of the vortex is being captured in this data. Capturing data from all directions with similar windspeeds will help to provide a deeper investigation into the passing vortices.

When evaluating the vertical wind velocity there is a strong harmonic at 3.33Hz which dissipates around 24Hz (Figure 3.13). This suggests a vertical circulation off the wing tip, similar to the investigation by Wei et. al., which they were able to map with PIV at 0.5D from the turbine (Figure 1.23). When plotting the horizontal vs. vertical components of the wind against each other, there is a circular quality that aligns with this concept (Figure 3.14). The two tightly clustered areas show a clear representation of the inflow with the vertical "stripes" proving a clear image of how the inflow has been distorted and ultimately creating the circular pattern over the 1,000-sample window with 30 blades passes by the CSAT. Although, this image does not provide a clear understanding of the timing or sequence in which these observations were made. A larger data set with the various prevailing winds and conditions will provide better insight to the formation and shedding of the vortices.

The ability to capture and quantify the inner-rotor fluid dynamics at a field scale can direct future VAWT research and design. The field scale data can be used to validate and develop numerical models that can provide a realistic look into the fluid-structure interactions, providing comprehension on the flow complexity experienced by each blade. With this understanding, VAWT design can be better understood while evaluating turbine parameters such as the solidity, aspect ratio, variable pitching, etc. and blade parameters such as airfoil shape, variable profile, edge serration, material, etc. Most of these which are mentioned in Barnes et al. standardization report (Barnes et al., 2021).

5. Conclusions and Future Work

A field scale research facility has been developed at the University of Minnesota Duluth Research and Field Studies Center with a 2.4kW VAWT and high-resolution data acquisition system. The placement of a Campbell Scientific CSAT3BH 3D sonic anemometer inside the rotor of the wind turbine provides a cutting-edge method to evaluate the inner-rotor fluid-structure interactions. Paring this with the three measurement stations along the span of the tower, atmospheric observations in the lower-level atmospheric boundary layer can be collected to support VAWT and other research disciplines. The motivation for the facility is derived from the increased energy demand in the global energy outlook with the high projections of the wind energy contributions. VAWTs can be a key contributor to the energy generation mix with technological advancements contributing to the use in distributed wind, array clustering and offshore energy applications.

In general, VAWTs have been studied thoroughly in CFD simulations and adequately studied in wind tunnel experiments, but there is a gap VAWT field scale research. This is critical when it comes to the Reynolds number dependency, with many of the CFD and wind tunnel studies below $Re=3*10^5$ which does not accurately represent real-world conditions. VAWTs also have a distinctive dynamic stall effect due to their omnidirectionality and a constantly changing angle of attack. Due to this, the blades are consistently encountering flow complications from the upstream blade. As a blade enters a stall, vortices are being generated from the blade, further complicating the flow. This phenomenon previously has not been investigated in any field scale studies.

A significant part of this project is the development of the experimental setup as highlighted below:

- 20m (65ft) easily lowered steel tower
- 2.4kW Semtive Neomi M with a 3m (10ft) diameter and blade length
- Meteorological stations at 6.1m (20ft), 12.2m (40ft) and 18.3m (60ft)
- Atmospheric conditions (temperature, barometric pressure, relative humidity) sampled at 1Hz
- Wind speed and direction sampled at 32Hz at each station
- Wind speed and direction sampled up to 100Hz inside the rotor of the turbine

The preliminary investigation discussed in this paper aims to provide insight to some of the potential of the research facility with:

- Sensor alignment for the quality assurance and control of the data with a Pearson Correlation Coefficient r-value > 0.9

- Atmospheric conditions at the site and the affect of the averaging period
- Inner-rotor observations at 100Hz capturing blade tip vortices

While the fundamental groundwork has been completed on the UMD wind turbine, there are several recommendations to improve the facility and many research opportunities available in the VAWT sector and other disciplines alike.

5.1. Power Performance

The operation of the turbine was limited throughout this study, resulting in an insufficient amount of data to characterize the performance of the turbine. After updates are done on the electrical controls, more data can be collected on the turbine performance characteristics, including the real-time power output to evaluate how various atmospheric conditions affect the power production. As the research articles in this paper mention, the power coefficient is often compared to the tip speed ratio of the turbine. The parameters can also be evaluated with other atmospheric conditions to classify the turbine performance. This will help with comparing the fluid-structure interactions to other CFD, wind tunnel and field scale studies.

5.2. Structural Reinforcement

The torque loading from the turbine produces vibrations in the wind turbine tower. This effect is amplified when a substantial wind gust occurs, and the tower is unloading. In smooth conditions amplified tower vibration and oscillations have not been witnessed. This can lead to fatigue on the equipment mounted on the tower. An investigation into the loading and tower oscillations should be conducted to see if more precautions, such as guy wires, should be implemented. The current boom arms are 2” aluminum pipe which are coupled together. Replacing these with steel pipe will help to increase the stiffness of the mounting system. These additional measures will increase the safety understanding of the site and could lead to better data collection, with less movement of the boom arms and sensors. With the reinforcement measures, the sensors alignment should be investigated as previously mentioned. While structural reinforcement may help with the quality control and reliability, applying a higher order filter to the data may be necessary as well.

5.3. Final Remarks

This project has resulted in a multifunctional state-of-the-art facility that will contribute to the development of VAWTs. The data can also be used in other research fields, teaching and community outreach. With the ease of accessibility to the location and the rigorous weather conditions experienced at the site, this experimental setup can provide insight to wind energy characteristics extreme climates, such as the arctic. The data acquisition system was designed with expansion in mind, which will allow future research to be easily adapted. The abilities of this facility are profound such as blade/tower strain, acoustics, vibratory, materials and the many other wind energy design challenges, which will all contribute to development for VAWTs in distributed energy, array clustering and offshore wind applications.

6. Bibliography

Alabdali, Q. A., Bajawi, A. M., Fatani, A. M., & Nahhas, A. M. (2020). Review of Recent Advances of Wind Energy. *Sustainable Energy, Vol. 8, 2020, Pages 12-19*, *8*(1), 12–19. https://doi.org/10.12691/RSE-8-1-3

Barnes, A., Marshall-Cross, D., & Hughes, B. R. (2021). Towards a standard approach for future Vertical Axis Wind Turbine aerodynamics research and development. *Renewable and Sustainable Energy Reviews*, *148*, 111221. https://doi.org/10.1016/J.RSER.2021.111221

Bianchini, A., Bangga, G., Baring-gould, I., & Croce, A. (2022). *Current status and grand challenges for small wind turbine technology*. *2*, 2003–2037.

Bogateanu, R., Dumitrache, A., Dumitrescu, H., & Stoica, C. I. (2014). Reynolds number efects on the aerodynamic performance of small VAWTs. *UPB Scientific Bulletin, Series D: Mechanical Engineering*, *76*(1), 25–36.

Brownstein, I. D., Kinzel, M., & Dabiri, J. O. (2016). Performance enhancement of downstream vertical-axis wind turbines. *JOURNAL OF RENEWABLE AND SUSTAINABLE ENERGY 8*. https://doi.org/10.1063/1.4964311

Carrigan, T. J., Dennis, B. H., Han, Z. X., & Wang, B. P. (2012). Aerodynamic Shape Optimization of a Vertical-Axis Wind Turbine Using Differential Evolution. *ISRN Renewable Energy*, *2012*, 1–16. https://doi.org/10.5402/2012/528418

Energy Agency, I. (2021). *World Energy Outlook 2021*. www.iea.org/weo

Energy Information Administration, U. (2022). *AEO2022 Narrative*. www.eia.gov

Ennis, B. L., & Griffith, D. T. (2018). *System Levelized Cost of Energy Analysis for Floating Offshore Vertical-Axis Wind Turbines. August*. http://www.osti.gov/servlets/purl/1466530/

Goit, J. P., Taguchi, M., Tatsuno, J., & Kameda, T. (2022). Effect of averaging time windows on wind resource assessment of small wind turbines. *Wind Energy*. https://doi.org/10.1002/WE.2723

Gorle, J. M. R., Chatellier, L., Pons, F., & Ba, M. (2019). Modulated circulation control around the blades of a vertical axis hydrokinetic turbine for flow control and improved performance. *Renewable and Sustainable Energy Reviews*, *105*, 363–377. https://doi.org/10.1016/J.RSER.2019.02.007

Gosselin, R., Dumas, G., & Boudreau, M. (2013, May). *(PDF) Parametric study of H-Darrieus vertical-axis turbines using uRANS simulations*. https://www.researchgate.net/publication/304805301_Parametric_study_of_H-Darrieus_vertical-axis_turbines_using_uRANS_simulations

Hand, B., Kelly, G., & Cashman, A. (2021). Aerodynamic design and performance parameters of a lift-type vertical axis wind turbine: A comprehensive review. *Renewable and Sustainable Energy Reviews*, *139*, 110699. https://doi.org/10.1016/J.RSER.2020.110699

Keijer, W. J. T. (2020). *3D aerodynamic effects and circulation control of vertical axis wind turbines (MSc thesis)*. *July*. http://resolver.tudelft.nl/uuid:765b00ba-7a80-4588-8476-774d7badc6cb

Kelvin Lwangu, L. (2020). *EXPERIMENTAL INVESTIGATION INTO THE INFLUENCE OF TURBULENCE ON THE PERFORMANCE OF A SMALL-SCALE VERTICAL AXIS WIND TURBINE IN A WIND TUNNEL ENVIRONMENT*.

Kinzel, M., Araya, D. B., & Dabiri, J. O. (2015). Turbulence in vertical axis wind turbine canopies. *Physics of Fluids*, *27*(11), 115102. https://doi.org/10.1063/1.4935111

Li, an, Maeda, T., Kamada, Y., Murata, J., Yamamoto, M., Ogasawara, T., Shimizu, K., & Kogaki, T. (2016). *Study on power performance for straight-bladed vertical axis wind turbine by field and wind tunnel test*. https://doi.org/10.1016/j.renene.2016.01.002

McIntosh, S. (2009). Wind energy for the built environment. *Built Environment*, *May*. https://doi.org/https://doi.org/10.17863/CAM.62306

Möllerström, E., Gipe, P., Beurskens, J., & Ottermo, F. (2019). A historical review of vertical axis wind turbines rated 100 kW and above. *Renewable and Sustainable Energy Reviews*, *105*, 1–13. https://doi.org/10.1016/J.RSER.2018.12.022

Möllerström, E., Ottermo, F., Goude, A., Eriksson, S., Hylander, J., & Bernhoff, H. (2016). Turbulence influence on wind energy extraction for a medium size vertical axis wind turbine. *Wind Energy*, *19*(11), 1963–1973. https://doi.org/10.1002/WE.1962

Musial, W., Spitsen, P., Duffy, P., Beiter, P., Marquis, M., Hammond, R., & Shields, M. (2022). *Offshore Wind Market Report: 2022 Edition; Executive Summary.* https://www.osti.gov/servlets/purl/1883382/

Óskarsdóttir, M. Ó. (2014). *A General Description and Comparison of Horizontal Axis Wind Turbines and Vertical Axis Wind Turbines*.

Pearson, C. (2013). *Vertical Axis Wind Turbine Acoustics*.

Sagharichi, A., Zamani, M., & Ghasemi, A. (2018). Effect of solidity on the performance of variable-pitch vertical axis wind turbine. *Energy*, *161*, 753–775. https://doi.org/10.1016/j.energy.2018.07.160

Shepherd, D. G. (1990). *Historical development of the windmill*. https://doi.org/10.2172/6342767

Sun, X., & Zhou, D. (2021). Review of Numerical and Experimental Studies on Flow Characteristics around A Straight-bladed Vertical Axis Wind Turbine and Its Performance Enhancement Strategies. *Archives of Computational Methods in Engineering*, *29*(3), 1839–1874. https://doi.org/10.1007/S11831-021-09640-4/FIGURES/38

Tummala, A., Velamati, R. K., Sinha, D. K., Indraja, V., & Krishna, V. H. (2016). A review on small scale wind turbines. *Renewable and Sustainable Energy Reviews*, *56*, 1351–1371. https://doi.org/10.1016/J.RSER.2015.12.027

Wei, N. J., Brownstein, I. D., Cardona, J. L., Howland, M. F., & Dabiri, J. O. (2019). *Near-wake structure of full-scale vertical-axis wind turbines*. https://doi.org/10.1017/jfm.2020.578

Wilhelm, J. P. (2010). *Power Envelope Expansion using a Solidity Matching Scheme for a Circulation Controlled Vertical Axis Wind Turbine*. http://hdl.handle.net/10450/11221

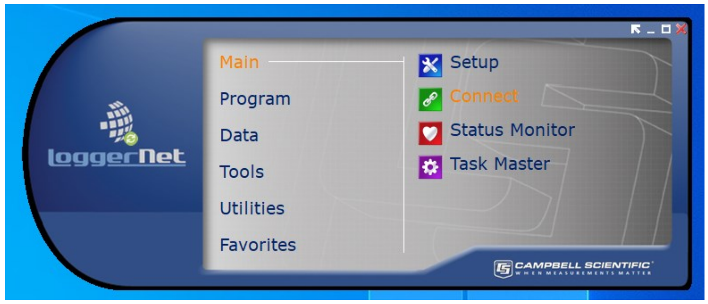
7. Appendices

Table 7.1: Guide to save the data from the CR1000X dataloggers.

Guide on saving data from defined data tables on datalogger program

1. Open LoggerNet from the desktop

2.Open Setup from Main menu



3.Under the Setup screen, select the data logger of interest.

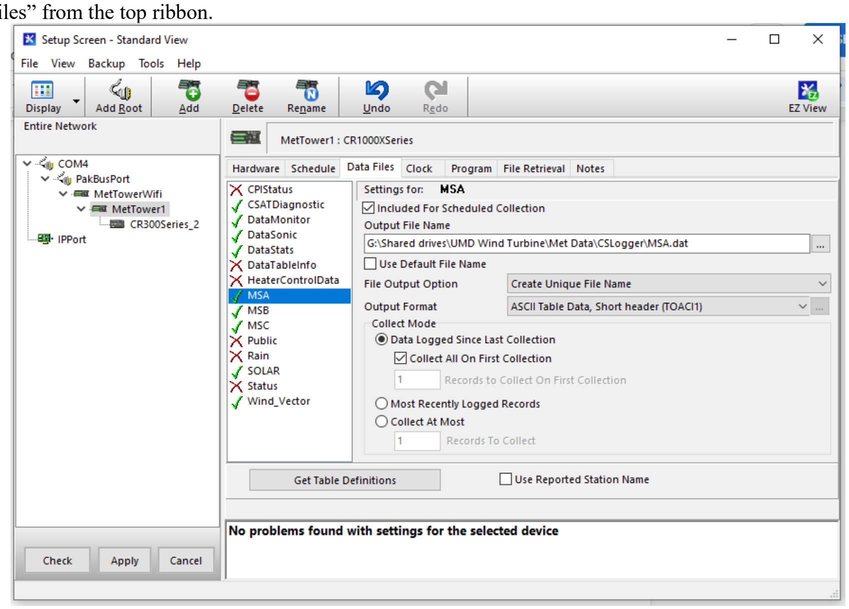
a. MetTower1: CR1000X data logger that is currently running the campbell scientific (CS) sensors

i. This includes the 3 METSENS, CSAT3BH, and CS320 pyranometer

b. MetTowerWifi: CR1000X data logger that is currently running the gill windmasters

i. This includes all 3 sampling at approximately 32 hz.

4.Select "Data Files" from the top ribbon.



5.On the left side are the defined data tables from the program that is currently running on the data logger. Select the data table of interest to change the settings.

a. "Include for Scheduled Collection" box will determine if the data will be collected from the data logger and stored locally. The "red X" or "green Check" will change next to the data table name on the list of defined data tables.

b. Adjust the settings as desired.

i. Output File Name: Location of where to save data tables locally. For low-resolution data (Metsens, Solar) save the data to G: drive for remote access.

ii. For high-resolution data (CSAT, WM) save data to the external hard drive.

1. .dat or .csv file will work for data importing (.dat is default)

2. DO NOT STORE DATA LOCALLY ON LAPTOP

iii. File Output Option: "Create Unique File Name" this adds a new file indexed with _yyyy_MM_DD in the directory

iv. Output Format: "ASCII Table Data, Long header (TOA5)" creates a header with multiple rows of information (Will need to specify header row #, etc for post processing) – Keep TOA5 for the timestamp to include milliseconds

v.Collect Mode: Settings as seen in figure below, have not encountered any issues with this method

6.The file retrieval tab is where to change how often data is retrieved. When switching data logger programs, this does not need to be checked (will not change).

Table 7.2: Guide to switch CRBasic programs on CR1000X dataloggers (all Campbell Scientific sensors to high-res).

Guide on uploading a new program to the datalogger

DAQ and Program File Names:

MetTower1: CR1000X data logger that is currently running the campbell scientific (CS) sensors

WindTurbineCS-1.2.CR1X :This includes the 3 METSENS, CSAT3BH, and CS320

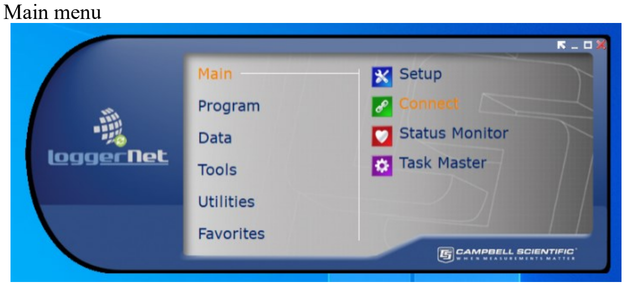
WindTurbineCS-CSAT100.CR1X :This is only the CSAT at 100Hz

MetTowerWifi: CR1000X data logger that is currently running the Gill WindMasters

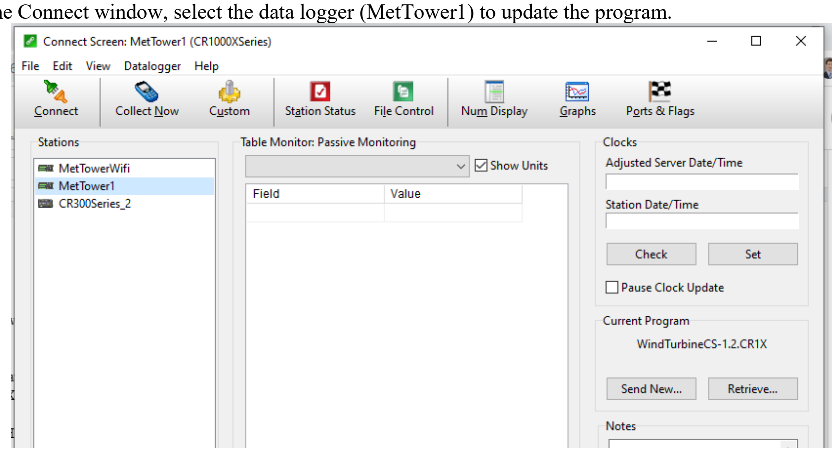
WindTurbineWM-2.0 :This includes all 3 sampling at 32 hz.

Procedure:

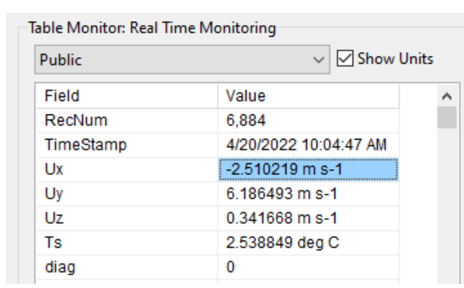
1. IMPORTANT: Before switching back to the standard DAQ programs, select Collect Now to download all of the data.
2. To run the CSAT3BH at 100 Hz, send a new program to MetTower1
3. The current program with all the CS sensors is WindTurbineCS-1.2.CR1X
4. The program to run the CSAT only (and thermocouple when working) at 100Hz is: WindTurbineCS-CSAT100.CR1X
   a. The windmasters will continue to sample at 32Hz if the program is not changed on MetTowerWifi
5. Open LoggerNet from the desktop
   a. Open Connect from Main menu



   b. From the Connect window, select the data logger (MetTower1) to update the program.



   c. Connect to the data logger with Connect button in top left corner
   d. Select Collect Now to download any data since the last files were written on the hour.
   e. Select Send New… under current program on right side
   f. Programs are in the UMD Wind Turbine shared G: drive
      i. Continue through any pop-up notifications.
      ii. Check the current program is running:
         1. On the Table Monitor drop down select public and verify the data values are changing for the desired sensor(s)



6. See Saving Data doc to verify the settings are correct
7. IMPORTANT: Before switching back to the standard DAQ programs, select Collect Now to download all of the high resolution data.

Easier way

Can also be switch by selecting “File Control” on the ribbon, selecting the desired program and “run now”

Table 7.3: CR1000X Datalogger CRBasic code to control all the Campbell Scientific sensors.

```
'Structure -- Constants/Variables/Units & Data Tables -- Program

'Current Devices                  Input
'CSAT3BH                          CPI/C7
'METSens500 A                     SDM-SIO4A 1 (32)
'METSens500 B                     SDM-SIO4A 2 (33)
'METSens500 C                     SDM-SIO4A 3 (34)
'CS320 Solar                      C5
'52202-L Rain Gauge               P1
'Thermocouple
PipeLineMode

Const FAST_SCAN_INTERVAL = 20 'Scan at 50Hz= 1/.02s '=10 for 100Hz    (CSAT3BH &
Thermocouple)
Const SCAN_INTERVAL = 1       'Scan at 1Hz   (METSens500)
Const MONITOR_INTERVAL = 5       'Scan at 0.2Hz (5sec) (Solar, Rain, Heater Control)

'CSAT3H
Const CPI_ADDR = 30        'CPI Address of Device
Const SDM_ADDR = 3
Const STATS_INTERVAL = 30  'Period of 30 min for online statistics
Const SDM_RATE = 30        'SDM Rate to 30
Const N_STATS = 1          'Increment size in the totalization functions
Const CSAT_AZIMUTH = 210     'Direction of compass convention in decimal degree that is
pointed by CSAT3BH.
Public wind(5)
Alias wind(1) = Ux 'x-axis wind      Variable names for alias wind(1-5) must not change
with heated sonic!
Alias wind(2) = Uy 'y-axis wind
Alias wind(3) = Uz 'z-axis wind
Alias wind(4) = Ts 'Air Temperature
Alias wind(5) = diag 'Diagnostics
Units Ux = m s-1
Units Uy = m s-1
Units Uz = m s-1
Units Ts = deg C

'CSAT3B monitor outputs
Public monitor(4)
Alias monitor(1) = board_temp 'Board Temperature
Alias monitor(2) = board_humidity 'Board Relative Humidity
Alias monitor(3) = incline_pitch 'Inclinometer Pitch
Alias monitor(4) = incline_roll 'Inclinometer Roll
Units board_temp = deg C
Units board_humidity = percent
Units incline_pitch = decimal deg
Units incline_roll = decimal deg
'CSAT3B diagnostic flags
Public diag_bits(9) As Boolean
Alias diag_bits(1) = Flag_LowAmp 'Low Amplitude Flag
Alias diag_bits(2) = Flag_HighAmp 'High Amplitude Flag
Alias diag_bits(3) = Flag_Tracking 'Tracking Flag
Alias diag_bits(4) = Flag_HighDeltaC 'High Delta Speed of Sound Flag
Alias diag_bits(5) = Flag_Acq 'Acquiring Flag
Alias diag_bits(6) = Flag_LowVolt 'Low Supply Voltage Flag
Alias diag_bits(7) = Flag_TrigErr 'Memory Error Flag
Alias diag_bits(8) = Flag_InternalRH 'The relative humidity inside of the CSAT3BH
electronics head is greater than 80%
Alias diag_bits(9) = Flag_MemErr 'Trigger Error Flag

'Wind directions and speed.
Public wind_spd_dir(5)
Alias wind_spd_dir(1) = wind_spd_mean       'Mean wind speed
Alias wind_spd_dir(2) = wind_spd_rsult      'Resultant  wind speed (AmeriFlux notation)
Alias wind_spd_dir(3) = wind_dir_csat       'Wind direction in the CSAT coordinate
system (i.e right-handed coordinate system).
Alias wind_spd_dir(4) = wind_dir_std        'Wind direction standard deviation
```

```
Alias wind_spd_dir(5) = wind_dir_compass    'Wind direction in the compass convention
Units wind_spd_mean    = m s-1
Units wind_spd_rsult   = m s-1
Units wind_dir_csat    = decimal deg
Units wind_dir_std     = decimal deg
Units wind_dir_compass = decimal deg

'Has to be in Logger program
'Variables from heater controller
Public com_rslt_get As Long     'Result code for communication to get variables
Units  com_rslt_get = adimensional

Public com_rslt_snt As Long     'Result code for communication to sent variables
Units  com_rslt_snt = adimensional

' Has to be in Logger program
Dim array_frm_heating_ctrl(17)  'Variable array output from CSAT3BH heater controller
Alias array_frm_heating_ctrl(1) = T_trnsd_uppr    'Upper transducer temperature
Alias array_frm_heating_ctrl(2) = T_trnsd_lwr         'Lower transducer temperature
Alias array_frm_heating_ctrl(3) = T_arm_uppr      'Upper arm temperature
Alias array_frm_heating_ctrl(4) = T_arm_lwr       'Lower arm temperature
Alias array_frm_heating_ctrl(5) = T_amb_ctrl      'Average ambient air temperature
(controller)
Alias array_frm_heating_ctrl(6) = RH_amb_ctrl     'Average ambient RH (controller)
Alias array_frm_heating_ctrl(7) = T_DP_amb_ctrl   'Average ambient dew point
(controller)
Alias array_frm_heating_ctrl(8) = pwr_trnsds      'Power through transducers for heating
Alias array_frm_heating_ctrl(9) = pwr_arms   'power through arms for heating
Alias array_frm_heating_ctrl(10)= pwr_trnsds_arms 'Heater power
Alias array_frm_heating_ctrl(11)= trnsd_heater_on 'Transducer heater on (-1)/off (0)
Alias array_frm_heating_ctrl(12)= arm_heater_on   'Arm heater on (-1)/off (0)
Alias array_frm_heating_ctrl(13)= trnsd_heater_fail   'Transducer heater fail (-
1)/normal (0)
Alias array_frm_heating_ctrl(14)= arm_heater_fail     'Arm heater fail (-1)/normal (0)
Alias array_frm_heating_ctrl(15)= trnsd_ice_wet_lck   'Seconds after sonic_amp_l_f,
_sig_lck_f, and/or _aq_sig_f are high while (T_amb_ctrl - T_DP_amb_ctrl) <
T_OFFST_TRNSDS
Alias array_frm_heating_ctrl(16)= heating_inaplcbl_f  'flag to indicate sonic heating
would be inapplicable to improve the measurements
Alias array_frm_heating_ctrl(17)= CSAT3H_ctrl_on  'Control status of CR300 in controller
is if "-1 and off if "0"

Units T_trnsd_uppr       = deg C
Units T_trnsd_lwr        = deg C
Units T_arm_uppr         = deg C
Units T_arm_lwr          = deg C
Units T_amb_ctrl         = deg C
Units RH_amb_ctrl        = %
Units T_DP_amb_ctrl      = deg C
Units pwr_arms           = W
Units pwr_trnsds         = W
Units pwr_trnsds_arms    = W
Units arm_heater_on      = adimensional
Units trnsd_heater_on    = adimensional
Units arm_heater_fail    = adimensional
Units trnsd_heater_fail  = adimensional
Units trnsd_ice_wet_lck  = s
Units heating_inaplcbl_f = adimensional
Units CSAT3H_ctrl_on     = adimensional

Dim pwr_hr_trnsds_arms
Units pwr_hr_trnsds_arms= WH

' Must be in logger program, part of heated sonic function
Dim   array_to_heating_ctrl(7) = {NaN, NaN, 0, 86, MONITOR_INTERVAL, FALSE, 0}
'Variable array from sonic measurements to CSAT3_H heater controller
Alias array_to_heating_ctrl(1) = U_total_rung_mean      'Total wind velocity of running
mean over a monitor interval
Alias array_to_heating_ctrl(2) = Ts_rung_mean           'Ts running mean over a monitor
interval
```

```
Alias array_to_heating_ctrl(3) = diag_sonic_slwintv     'Aggregated diagnosis code over
a monitor interval
Alias array_to_heating_ctrl(4) = press_rung_mean        'press running mean over a
monitor interval
Alias array_to_heating_ctrl(5) = com_intv               'Interval of communication
between main Datalogger and CR300 inside the heating controller
Alias array_to_heating_ctrl(6) = CSAT3H_user_ctrl_on    'Control heating for CSAT3BH by
user from main Datalogger (CSAT3H_user_ctrl_on = - CSAT3H_user_ctrl_flg)
Alias array_to_heating_ctrl(7) = rst_secs_snc_last_com  'Used to reset
secs_snc_last_com as zero in the heating controller every communication
Units U_total_rung_mean     = m s-1
Units Ts_rung_mean          = Deg C
Units diag_sonic_slwintv    = adimensional
Units press_rung_mean       = kPa
Units com_intv              = s
Units CSAT3H_user_ctrl_on   = adimensional
Units rst_secs_snc_last_com = s

Dim    Diag_sonic As Long
Dim    U_total                                   'Total wind in a main scan
Public CSAT3H_user_ctrl_flg As Boolean = TRUE         'Used to change
CSAT3H_user_ctrl_on while program is running

'Working variables.
Dim diag_integer As Long
Dim disable_stats(2) As Boolean 'Intermediate processing disable flags
Dim i As Long

'Other Variables
Public battery_volt
Units battery_volt = V
Public panel_temp
Units panel_temp = deg C
' Original Sonic Data
DataTable (DataSonic,1,-1)
  DataInterval (0,FAST_SCAN_INTERVAL,mSec,0) '50Hz
  TableFile ("USB:D:\CSData\CSAT\DS",8,-1,0,1,Hr,0,0) 'New table each hour
  Sample (5,wind(1),IEEE4)
  Sample (1,wind_dir_compass,IEEE4)
  Sample (1,wind_dir_csat,IEEE4)
  Sample (1,wind_spd_mean,IEEE4)
  Sample (1,wind_spd_rsult,IEEE4)
EndTable
' Original Sonic Data
DataTable (CSATDiagnostic,1,-1)
  DataInterval (0,MONITOR_INTERVAL,Sec,0) '50Hz
  TableFile ("USB:D:\CSData\CSAT\CSATDiagnotstic",8,-1,0,1,Hr,0,0) 'New table each hour
  Sample (5,wind(1),IEEE4)
  Sample (1,Diag_sonic,Long)
  Sample (7,array_to_heating_ctrl(),IEEE4)
  Totalize (1,N_STATS,IEEE4,disable_stats(1))
  FieldNames("csat_normal")
  Totalize (1,N_STATS,IEEE4,NOT (disable_stats(1) OR disable_stats(2)))
  FieldNames ("csat_warnings")
  Totalize (1,N_STATS,IEEE4,NOT (Flag_LowAmp) OR NOT (disable_stats(2)))
  FieldNames ("Flag_LowAmp_Tot")
  Totalize (1,N_STATS,IEEE4,NOT (Flag_HighAmp) OR NOT (disable_stats(2)))
  FieldNames ("Flag_HighAmp_Tot")
  Totalize (1,N_STATS,IEEE4,NOT (Flag_Tracking) OR NOT (disable_stats(2)))
  FieldNames ("Flag_Tracking_Tot")
  Totalize (1,N_STATS,IEEE4,NOT (Flag_HighDeltaC) OR NOT (disable_stats(2)))
  FieldNames ("Flag_HighDeltaC_Tot")
  Totalize (1,N_STATS,IEEE4,NOT (Flag_Acq) OR NOT (disable_stats(2)))
  FieldNames ("Flag_Acq_Tot")
  Totalize (1,N_STATS,IEEE4,NOT (Flag_LowVolt) OR NOT (disable_stats(2)))
  FieldNames ("Flag_LowVolt_Tot")
  Totalize (1,N_STATS,IEEE4,NOT (Flag_MemErr) OR NOT (disable_stats(2)))
  FieldNames ("Flag_TrigErr_Tot")
  Totalize (1,N_STATS,IEEE4,NOT (Flag_InternalRH) OR NOT (disable_stats(2)))
  FieldNames ("Flag_InternalRH_Tot")
```

```
  Totalize (1,N_STATS,IEEE4,NOT (Flag_TrigErr) OR NOT (disable_stats(2)))
  FieldNames ("Flag_MemErr_Tot")
EndTable

'Monitor Data
DataTable (DataMonitor,1,-1)
  DataInterval (0,MONITOR_INTERVAL,Sec,0) '5 sec (0.2 Hz)
  TableFile ("USB:D:\CSData\CSAT\DataMonitor",8,-1,0,1,Hr,0,0) 'New table each hour
  Sample (4,monitor(1),IEEE4)
EndTable

'Compute wind direction from CSAT data.
DataTable (Wind_Vector, TRUE,-1)
  DataInterval (0, FAST_SCAN_INTERVAL ,mSec, 0) '32Hz -- not needed if post processing?
  TableFile ("USB:D:\CSData\CSAT\DSWV",8,-1,0,1,Hr,0,0) 'New table each hour
  WindVector   (1, Uy, Ux, IEEE4, disable_stats(1), 0, 1, 2)
EndTable

 'Statistics data.
DataTable (DataStats,TRUE,-1)
  DataInterval (0,FAST_SCAN_INTERVAL,Sec,0) '30 min
  TableFile ("USB:D:\CSData\CSAT\DSStats",8,-1,0,1,Hr,0,0) 'new table every 4 hrs
  Average (1,Ts,IEEE4,disable_stats(1))
  StdDev (1,Ts,IEEE4,disable_stats(1))
  Average (1,Ux,IEEE4,disable_stats(1))
  StdDev (1,Ux,IEEE4,disable_stats(1))
  Average (1,Uy,IEEE4,disable_stats(1))
  StdDev (1,Uy,IEEE4,disable_stats(1))
  Average (1,Uz,IEEE4,disable_stats(1))
  StdDev (1,Uz,IEEE4,disable_stats(1))
  Sample (1,wind_dir_compass,IEEE4)
  Sample (1,wind_dir_csat,IEEE4)
  Sample (1,wind_spd_mean,IEEE4)
  Sample (1,wind_spd_rsult,IEEE4)
  Sample (1,wind_dir_std,IEEE4)
  Totalize (1,N_STATS,IEEE4,disable_stats(1))
  FieldNames("csat_normal")
  Totalize (1,N_STATS,IEEE4,NOT (disable_stats(1) OR disable_stats(2)))
  FieldNames ("csat_warnings")
  Totalize (1,N_STATS,IEEE4,NOT (Flag_LowAmp) OR NOT (disable_stats(2)))
  FieldNames ("Flag_LowAmp_Tot")
  Totalize (1,N_STATS,IEEE4,NOT (Flag_HighAmp) OR NOT (disable_stats(2)))
  FieldNames ("Flag_HighAmp_Tot")
  Totalize (1,N_STATS,IEEE4,NOT (Flag_Tracking) OR NOT (disable_stats(2)))
  FieldNames ("Flag_Tracking_Tot")
  Totalize (1,N_STATS,IEEE4,NOT (Flag_HighDeltaC) OR NOT (disable_stats(2)))
  FieldNames ("Flag_HighDeltaC_Tot")
  Totalize (1,N_STATS,IEEE4,NOT (Flag_Acq) OR NOT (disable_stats(2)))
  FieldNames ("Flag_Acq_Tot")
  Totalize (1,N_STATS,IEEE4,NOT (Flag_LowVolt) OR NOT (disable_stats(2)))
  FieldNames ("Flag_LowVolt_Tot")
  Totalize (1,N_STATS,IEEE4,NOT (Flag_MemErr) OR NOT (disable_stats(2)))
  FieldNames ("Flag_TrigErr_Tot")
  Totalize (1,N_STATS,IEEE4,NOT (Flag_InternalRH) OR NOT (disable_stats(2)))
  FieldNames ("Flag_InternalRH_Tot")
  Totalize (1,N_STATS,IEEE4,NOT (Flag_TrigErr) OR NOT (disable_stats(2)))
  FieldNames ("Flag_MemErr_Tot")
  Average (1,panel_temp,IEEE4,FALSE)
  Average (1,battery_volt,IEEE4,FALSE)
  Average (1,board_temp,IEEE4,FALSE)
  Average (1,board_humidity,IEEE4,FALSE)
  Average (1,incline_pitch,IEEE4,FALSE)
  Average (1,incline_roll,IEEE4,FALSE)
EndTable

'Heater Controller Data
DataTable (HeaterControlData,1,-1)
  DataInterval (0,MONITOR_INTERVAL,Min,0) 'samples every 5 min
  TableFile ("USB:D:\CSData\CSAT\HCD",8,-1,0,24,Hr,0,0) 'creates new table every 4
hours
```

```
  Sample (17,array_frm_heating_ctrl(),IEEE4,FALSE)
EndTable

'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'METSENS500 A
Public MSDataA(7)
Public MetSENS_StatusA As Long
Public CommResultA(3) As Long
'Data for METSENS500 A
Alias MSDataA(1)=WindDirA
Alias MSDataA(2)=WS_msA
Alias MSDataA(3)=Mag_HeadA
Alias MSDataA(4)=BP_mbarA
Alias MSDataA(5)=RHA
Alias MSDataA(6)=AirT_CA
Alias MSDataA(7)=DP_CA

Units Mag_HeadA=degrees
Units MetSENS_StatusA=code
Units BP_mbarA=mbar
Units RHA=%
Units DP_CA=Deg C
Units WindDirA=degrees
Units WS_msA=meters/second
Units AirT_CA=Deg C

 DataTable(MSA,True,-1)
  DataInterval(0,SCAN_INTERVAL,Sec,0) '10 Hz sampling
  TableFile ("USB:G:\Shared drives\UMD Wind Turbine\Met Data\CSLogger\MSA",8,-
1,0,1,Hr,0,0)
     Sample(1,BP_mbarA,IEEE4)
     Sample(1,RHA,FP2)
     Sample(1,AirT_CA,FP2)
     Sample(1,DP_CA,FP2)
     Sample(1,WindDirA,FP2)
     Sample(1,WS_msA,FP2)
     Sample(1,MetSENS_StatusA,FP2)
     Sample(1,CommResultA(1),FP2)
     Sample(1,Mag_HeadA,FP2)
 EndTable

'METSENS500 B
Public MSDataB(7)
Public MetSENS_StatusB As Long
Public CommResultB(3) As Long
'Data for METSENS500 A
Alias MSDataB(1)=WindDirB
Alias MSDataB(2)=WS_msB
Alias MSDataB(3)=Mag_HeadB
Alias MSDataB(4)=BP_mbarB
Alias MSDataB(5)=RHB
Alias MSDataB(6)=AirT_CB
Alias MSDataB(7)=DP_CB

Units Mag_HeadB=degrees
Units MetSENS_StatusB=code
Units BP_mbarB=mbar
Units RHB=%
Units DP_CB=Deg C
Units WindDirB=degrees
Units WS_msB=meters/second
Units AirT_CB=Deg C


 DataTable(MSB,True,-1)
  DataInterval(0,SCAN_INTERVAL,Sec,0) '10 Hz sampling
  TableFile ("USB:G:\Shared drives\UMD Wind Turbine\Met Data\CSLogger\MSB",8,-
1,0,1,Hr,0,0)
     Sample(1,BP_mbarB,IEEE4)
     Sample(1,RHB,FP2)
```

```
    Sample(1,AirT_CB,FP2)
    Sample(1,DP_CB,FP2)
    Sample(1,WindDirB,FP2)
    Sample(1,WS_msB,FP2)
    Sample(1,MetSENS_StatusB,FP2)
    Sample(1,CommResultB(1),FP2)
    Sample(1,Mag_HeadB,FP2)
 EndTable

'METSENS500 C
Public MSDataC(7)
Public MetSENS_StatusC As Long
Public CommResultC(1) As Long
'Data for METSENS500 A
Alias MSDataC(1)=WindDirC
Alias MSDataC(2)=WS_msC
Alias MSDataC(3)=Mag_HeadC
Alias MSDataC(4)=BP_mbarC
Alias MSDataC(5)=RHC
Alias MSDataC(6)=AirT_CC
Alias MSDataC(7)=DP_CC

Units Mag_HeadC=degrees
Units MetSENS_StatusC=code
Units BP_mbarC=mbar
Units RHC=%
Units DP_CC=Deg C
Units WindDirC=degrees
Units WS_msC=meters/second
Units AirT_CC =Deg C

 DataTable(MSC,True,-1)
  DataInterval(0,SCAN_INTERVAL,Sec,0) '1 Hz sampling
  TableFile ("USB:G:\Shared drives\UMD Wind Turbine\Met Data\CSLogger\MSC",8,-
1,0,1,Hr,0,0)
  Sample(1,BP_mbarC,IEEE4)
    Sample(1,RHC,FP2)
    Sample(1,AirT_CC,FP2)
    Sample(1,DP_CC,FP2)
    Sample(1,WindDirC,FP2)
    Sample(1,WS_msC,FP2)
    Sample(1,MetSENS_StatusC,FP2)
    Sample(1,CommResultC(1),FP2)
    Sample(1,Mag_HeadC,FP2)
 EndTable
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'GPS
'Data for GPS
'Dim NMEASent(2) As String * 100
'Public BattV
'Public PTemp_C
'Public GPSData(15)
'Alias GPSData(1)=Latitude_A
'Alias GPSData(2)=Latitude_B
'Alias GPSData(3)=Longitude_A
'Alias GPSData(4)=Longitude_B
'Alias GPSData(5)=Speed
'Alias GPSData(6)=Course
'Alias GPSData(7)=MagVar
'Alias GPSData(8)=FixQual
'Alias GPSData(9)=NumSats
'Alias GPSData(10)=Altitude
'Alias GPSData(11)=PPS
'Alias GPSData(12)=SecSinceGPRMC
'Alias GPSData(13)=GPSReady
'Alias GPSData(14)=MaxClockChange
'Alias GPSData(15)=NumClockChange

'Units Latitude_A=degrees
'Units Latitude_B=minutes
```

```
'Units Longitude_A=degrees
'Units Longitude_B=minutes
'Units Speed=knots
'Units Course=degrees
'Units MagVar=degrees
'Units FixQual=code
'Units NumSats=count
'Units Altitude=meters
'Units PPS=microseconds
'Units SecSinceGPRMC=seconds
'Units GPSReady=count
'Units MaxClockChange=milliseconds
'Units NumClockChange=count

'DataTable(GPS1,True,-1)
'    DataInterval(0,MONITOR_INTERVAL,Sec,10) 'new table every 5 sec
' TableFile(GPS,0,-1,0,1,Hr,0,0) 'New table each hour
'    Minimum(1,BattV,FP2,False,False)
'    Sample(1,NMEASent(2),String)
'    Sample(1,Latitude_A,FP2)
'    Sample(1,Latitude_B,FP2)
'    Sample(1,Longitude_A,FP2)
'    Sample(1,Longitude_B,FP2)
'    Sample(1,Speed,FP2)
'    Sample(1,Course,FP2)
'    Sample(1,MagVar,FP2)
'    Sample(1,FixQual,FP2)
'    Sample(1,NumSats,FP2)
'    Sample(1,Altitude,FP2)
'    Sample(1,PPS,FP2)
'    Sample(1,SecSinceGPRMC,FP2)
'    Sample(1,GPSReady,FP2)
'    Sample(1,MaxClockChange,FP2)
'    Sample(1,NumClockChange,FP2)
'    Sample(1,NMEASent(1),String)
'EndTable

'RAIN GUAGE
'Public Rain_mm
'Units Rain_mm=mm
'
'DataTable(Rain,True,-1)
'    DataInterval(0,1,Min,10) '
'  TableFile (Rain,0,-1,0,1,Hr,0,0) 'New table each hour
'    Totalize(1,Rain_mm,FP2,False)
'EndTable

'SOLAR RADIATION
Const UVScanrate = MONITOR_INTERVAL
Public CS320(6)
Public SlrMJ
Alias CS320(1)=SlrRad_W
Alias CS320(2)=Raw_mV
Alias CS320(3)=SensorTemp
Alias CS320(4)=X_orient
Alias CS320(5)=Y_orient
Alias CS320(6)=Z_orient
Units SlrRad_W=W/M^2
Units Raw_mV=mV
Units SensorTemp=C
Units X_orient=deg
Units Y_orient=deg
Units Z_orient=deg
Units SlrMJ=MJ
'Define Data Tables
DataTable(SOLAR,True,-1)
  DataInterval(0,MONITOR_INTERVAL,Sec,10) '0.2Hz
  TableFile ("USB:D:\CSData\SOLAR\SOLAR",8,-1,0,1,Hr,0,0) 'New table each hour
  Average(1,SlrRad_W,FP2,False)
  Average(1,Raw_mV,FP2,False)
```

```
  Average(1,SensorTemp,FP2,False)
  Sample(1,X_orient,FP2)
  Sample(1,Y_orient,FP2)
  Sample(1,Z_orient,FP2)
EndTable

'Thermocouple and reference thermistor
'Declare Variables and Units
'Public BattV
'Public PTemp_C
'Public T107_C
'Public Temp_C
'
'Units BattV=Volts
'Units PTemp_C=Deg C
'Units T107_C=Deg C
'Units Temp_C=Deg C

'Define Data Tables
'DataTable(Thermocouple,True,-1)
'    DataInterval(0,FAST_SCAN_INTERVAL,mSec,10)
' TableFile (Thermocouple,0,-1,0,1,Hr,0,0)
'    Sample(1,T107_C,FP2)
'    Sample(1,Temp_C,FP2)
'EndTable

BeginProg
    'Open METSENS Ports on SDM-SIO4A (C1/C2/C3)
    SerialOpen(32,19200,50,50,5000)
    SerialOpen(33,19200,50,50,5000)
         SerialOpen(34,19200,50,50,5000)
  Scan (FAST_SCAN_INTERVAL,mSec,200,0)
'        'RAIN
'        PulseCount(Rain_mm,1,P1,1,0,0.1,0) '52202 Rain Gauge measurement 'Rain_mm'
'        CallTable Rain
    'CSAT3BH
    SerialOpen(ComC7,115200,4,0,500,3)      'opening for bidirectional communications
    press_rung_mean = 100
    PanelTemp (panel_temp,250) 'Measure data logger panel temperature
    Battery (battery_volt) 'Measure battery voltage.
    CSAT3H_user_ctrl_on = CSAT3H_user_ctrl_flg    'CSAT3H_user_ctrl_flg can be turn
on/off in public table
    'CSAT3B(Destination, Bus, Address, OperatingMode)
    CSAT3B(wind(),1,CPI_ADDR,0)
    'Break up the warning flags into separate bits.
    diag_integer = diag
    For i = 1 To 9
    diag_bits(i) = diag_integer AND &b000000001
    diag_integer = diag_integer >> 1
    Next i
    'Turn on this intermediate processing disable flag when any
    'warning flag is set or the data returned is NaN
    disable_stats(1) = ((diag_integer AND &b111111111) OR (diag = NaN))
    'Turn on this intermediate processing disable flag only when
    'CSAT3B diagnostic warning flags are set
    disable_stats(2) = ((diag_integer AND &b111111111) AND NOT (diag = NaN))
    'Variables to CSAT3BH heating controller
    AvgRun (Ts_rung_mean, 1, wind(4), (1000*MONITOR_INTERVAL/SCAN_INTERVAL))
'Running average of Ts
    U_total = SQR(Ux*Ux + Uy*Uy + Uz*Uz)
    AvgRun (U_total_rung_mean, 1, U_total, (1000*MONITOR_INTERVAL/SCAN_INTERVAL))
'Running average of press
    Diag_sonic        = IIF ((wind(5) <> NaN) AND (wind(5) <> -1), wind(5), &h3f)
    diag_sonic_slwintv = diag_sonic_slwintv OR Diag_sonic
    'Get the online wind vector statistics.
    CallTable Wind_Vector
    If (Wind_Vector.Output(1,1)) Then
      GetRecord (wind_spd_dir(1), Wind_Vector, 1)
      'Comput wind direction for compass convention
      wind_dir_compass = (360 + CSAT_AZIMUTH - wind_dir_csat) MOD 360
```

```
    EndIf
    'Working Variables
    CallTable DataSonic
    CallTable CSATDiagnostic
    CallTable DataStats
'    Thermocouple
'    'Default CR1000X Datalogger Battery Voltage measurement 'BattV'
'        Battery(BattV)
'        'Default CR1000X Datalogger Wiring Panel Temperature measurement 'PTemp_C'
'        PanelTemp(PTemp_C,15000)
'        '107 Temperature Probe measurement 'T107_C'
'        Therm107(T107_C,1,1,VX1,0,15000,1,0)
'        'Type E Thermocouple measurements 'Temp_C'
'        TCDiff(Temp_C,1,mv200C,2,TypeE,T107_C,True,500,15000,1,0)
'        'Call Data Tables and Store Data
'        CallTable Thermocouple
'  NextSubScan
  NextScan

  SlowSequence
  Scan(SCAN_INTERVAL,Sec,100,0)
'    METSENS500 A
'        ModbusMaster(CommResultA(1),32,19200,41,3,MSDataA(1),7,4,1,20,2) 'Atm. Press,
Rel. Humidity, Air Temp, Dew Pt.
        ModbusMaster(CommResultA(1),32,19200,41,3,MSDataA(1),1,7,1,20,2) 'Wind Speed
and Direction
'        ModbusMaster(CommResultA(3),32,19200,41,3,MetSENS_StatusA,25,1,3,100,2)
'Status
        CallTable MSA
'    METSENS500 B
'        ModbusMaster(CommResultB(1),33,19200,41,3,MSDataB(1),7,4,1,20,2) 'Atm. Press,
Rel. Humidity, Air Temp, Dew Pt.
        ModbusMaster(CommResultB(1),33,19200,41,3,MSDataB(1),1,7,1,20,2) 'Wind Speed
and Direction
'        ModbusMaster(CommResultB(3),33,19200,41,3,MetSENS_StatusB,25,1,3,100,2)
'Status
        CallTable MSB
'    METSENS500 C
'        ModbusMaster(CommResultC(1),34,19200,41,3,MSDataC(1),7,4,1,20,2) 'Atm. Press,
Rel. Humidity, Air Temp, Dew Pt.
        ModbusMaster(CommResultC(1),34,19200,41,3,MSDataC(1),1,7,1,20,2) 'Wind Speed
and Direction
'        ModbusMaster(CommResultC(3),34,19200,41,3,MetSENS_StatusC,25,1,3,100,2)
'Status
        CallTable MSC
NextScan
SlowSequence
Scan(MONITOR_INTERVAL,Sec,50,0)
'    UV
    SDI12Recorder(CS320(),C5,"0","M4!",1,0,-1)
    If SlrRad_W < 0 Then SlrRad_W = 0
    SlrMJ = SlrRad_W * UVScanrate * 1E-6
        CallTable SOLAR
'    GPS
'    Battery(BattV) 'Default CR1000X Datalogger Battery Voltage measurement 'BattV'
'       PanelTemp(PTemp_C,60) 'Default CR1000X Datalogger Wiring Panel Temperature
measurement 'PTemp_C'
'    GPS(GPSData(),-ComC7,0*3600,500,NMEASent())
'    CallTable GPS1
'    CSAT Heater
    'CSAT3BMonitor(Destination, Bus, Address)
    CSAT3BMonitor(monitor(),0,SDM_ADDR)
    CallTable(DataMonitor)
    GetVariables
(com_rslt_get,ComC7,0,2,0000,0,"Public","array_to_csat(1)",array_frm_heating_ctrl(1),17
)  'Has to be in Logger program
    If (com_rslt_get = 0) Then
     AvgRun (pwr_hr_trnsds_arms, 1, pwr_trnsds_arms, INT(60*60/MONITOR_INTERVAL))
    EndIf
```

```
    SendVariables
(com_rslt_snt,ComC7,0,2,0000,0,"Public","array_frm_csat(1)",array_to_heating_ctrl(1),7)
'Has to be in Logger program
    If (com_rslt_snt = 0) Then
     diag_sonic_slwintv = 0
    EndIf
    CallTable(HeaterControlData)
  NextScan
EndProg
```

Table 7.4: CR1000X Datalogger CRBasic code to operate the CSAT3BH at 100Hz.

```
'Structure -- Constants/Variables/Units & Data Tables -- Program

'Current Devices                  Input
'CSAT3BH                          CPI
'Thermocouple
PipeLineMode

Const FAST_SCAN_INTERVAL = 10 '=10 for 100Hz
Const CPI_ADDR = 30           'CPI Address of Device
Const SDM_ADDR = 3
Public wind(5)
Alias wind(1) = Ux 'x-axis wind      Variable names for alias wind(1-5) must not change
with heated sonic!
Alias wind(2) = Uy 'y-axis wind
Alias wind(3) = Uz 'z-axis wind
Alias wind(4) = Ts 'Air Temperature
Alias wind(5) = diag 'Diagnostics
Units Ux = m s-1
Units Uy = m s-1
Units Uz = m s-1
Units Ts = deg C

' Original Sonic Data
DataTable (DataSonic,1,-1)
  DataInterval (0,FAST_SCAN_INTERVAL,mSec,0) '100Hz
  TableFile ("USB:D:\CSData\CSAT\DS",0,-1,0,1,Hr,0,0) 'New table each hour
  Sample (5,wind(1),IEEE4)
EndTable

''Thermocouple and reference thermistor ' Uncomment to activate
''Declare Variables and Units
'Public BattV
'Public PTemp_C
'Public T107_C
'Public Temp_C
'Units BattV=Volts
'Units PTemp_C=Deg C
'Units T107_C=Deg C
'Units Temp_C=Deg C
'
''Define Data Tables
'DataTable(Thermocouple,True,-1)
'    DataInterval(0,FAST_SCAN_INTERVAL,mSec,10)
'    Sample(1,T107_C,FP2)
'    Sample(1,Temp_C,FP2)
'EndTable

BeginProg
  Scan (FAST_SCAN_INTERVAL,mSec,400,0)
'     'CSAT3BH
    CSAT3B(wind(),1,CPI_ADDR,0)
    CallTable DataSonic
'     'Thermocouple 'Uncomment to Activate
'     'Default CR1000X Datalogger Battery Voltage measurement 'BattV'
'         Battery(BattV)
'         'Default CR1000X Datalogger Wiring Panel Temperature measurement 'PTemp_C'
'         PanelTemp(PTemp_C,15000)
```

```
'        '107 Temperature Probe measurement 'T107_C'
'        Therm107(T107_C,1,1,VX1,0,15000,1,0)
'        'Type E Thermocouple measurements 'Temp_C'
'        TCDiff(Temp_C,1,mv200C,2,TypeE,T107_C,True,500,15000,1,0)
'        'Call Data Tables and Store Data
'        CallTable Thermocouple
  NextScan
EndProg

'Sonic Data
DataTable (DataSonic,1,-1)
  DataInterval (0,FAST_SCAN_INTERVAL,mSec,0) '50Hz
'  TableFile ("CRD:CS_DataSonic",64,-1,0,1,Hr,0,0) 'New table each hour
  Sample (5,wind(1),IEEE4)
EndTable
''Compute wind direction from CSAT data.
DataTable (Wind_Vector, TRUE,-1)
  DataInterval (0, FAST_SCAN_INTERVAL ,mSec, 0) '32Hz -- not needed if post processing?
'  TableFile ("CRD:CS_WindVector",64,-1,0,1,Hr,0,0) 'New table each hour
  WindVector   (1, Uy, Ux, IEEE4, disable_stats(1), 0, 1, 2)
EndTable

''Thermocouple and reference thermistor
''Declare Variables and Units
''Public BattV
''Public PTemp_C
''Public T107_C
''Public Temp_C
''
''Units BattV=Volts
''Units PTemp_C=Deg C
''Units T107_C=Deg C
''Units Temp_C=Deg C
'
''Define Data Tables
''DataTable(Thermocouple,True,-1)
''   DataInterval(0,FAST_SCAN_INTERVAL,mSec,10)
''   Sample(1,T107_C,FP2)
''   Sample(1,Temp_C,FP2)
''EndTable

BeginProg
  Scan (FAST_SCAN_INTERVAL,mSec,400,0)
    'CSAT3BH
    SerialOpen(ComC7,115200,4,0,500,3)      'opening for bidirectional communications
    PanelTemp (panel_temp,250) 'Measure data logger panel temperature
    Battery (battery_volt) 'Measure battery voltage
    'CSAT3B(Destination, Bus, Address, OperatingMode)
    CSAT3B(wind(),1,CPI_ADDR,0)
'    AvgRun (Ts_rung_mean, 1, wind(4), (1000*MONITOR_INTERVAL/SCAN_INTERVAL))
'Running average of Ts
    U_total = SQR(Ux*Ux + Uy*Uy + Uz*Uz)
'    AvgRun (U_total_rung_mean, 1, U_total, (1000*MONITOR_INTERVAL/SCAN_INTERVAL))
'Running average of press
    Diag_sonic        = IIF ((wind(5) <> NaN) AND (wind(5) <> -1), wind(5), &h3f)
    CallTable Wind_Vector
    If (Wind_Vector.Output(1,1)) Then
      GetRecord (wind_spd_dir(1), Wind_Vector, 1)
      'Comput wind direction for compass convention
      wind_dir_compass = (360 + CSAT_AZIMUTH - wind_dir_csat) MOD 360
    EndIf
    CallTable DataSonic
'    Thermocouple
'    'Default CR1000X Datalogger Battery Voltage measurement 'BattV'
'        Battery(BattV)
'        'Default CR1000X Datalogger Wiring Panel Temperature measurement 'PTemp_C'
'        PanelTemp(PTemp_C,15000)
'        '107 Temperature Probe measurement 'T107_C'
'        Therm107(T107_C,1,1,VX1,0,15000,1,0)
'        'Type E Thermocouple measurements 'Temp_C'
```

```
'        TCDiff(Temp_C,1,mv200C,2,TypeE,T107_C,True,500,15000,1,0)
'        'Call Data Tables and Store Data
'        CallTable Thermocouple
  NextScan
EndProg
```

Table 7.5: CR1000X Datalogger CRBasic code to scan with the three Gill WindMaster Pro's at 32Hz.

```
'CR1000X Series Datalogger
'Copyright (c) 2020 Campbell Scientific, Inc. All rights reserved.
'version 1.2

'SET PARAMETERS
'M1  continuous UVW
'U1  m/s
'O2  fixed field, comma delimited output
'L1  <CR><LF> record terminator
'P9  32 Hz measurement and output frequency
'B5 38.4 kbaud
'H2  no power up message
'NQ  address set to "Q"
'E1  Auto serial interface detection (signal red to +12Vdc is RS-232 and to G is RS-422)
'T1  analog output type, 0V to 5V (DNA)
'S1  analog range, full range is 5 m/s (DNA)
'C2  analog polar direction wrap around 0 to 360 deg (DNA)
'A3  Select sonic temperature output
'I1  Enable/disable analogue input message (DNA)
'J2  Set the unit resolution to high
'V1  Enable/disable PRT (DNA)
'X1  Not documented
'G0  Set online averaging to no averaging
'K50 Set Minimum direction velocity (DNA)

'Datalogger SDM
'SDM-C1  SDM-SIO4 SDM C1 (green)
'SDM-C2  SDM-SIO4 SDM C2 (white)
'SDM-C3  SDM-SIO4 SDM C3 (brown)
'G       SDM-SIO4 G (black)
'        SDM-SIO4 shield (clear)
'12V     SDM-SIO4 +12V (red)

'*** Wiring RS-232 ***

'CONTROL PORTS (RS-232 SERIAL)
'C5 (COM3 TX)         WindMaster Pro RxD (green)
'C6 (COM3 RX)         WindMaster Pro TxD (white)
'G                    WindMaster Pro Comms Ref (black)
'                     WindMaster Pro shield (clear)

'POWER OUT
'12V                  WindMaster Pro RS-232 Serial Mode (red on signal cable)

'*** Wiring RS-422 ***

'CONTROL PORTS (RS-232 SERIAL)
'C5 (COM3 TX)         SC110 DTE RxD (brown)
'C6 (COM3 RX)         SC110 DTE TxD (white)
'G                    SC110 DTE Ref (yellow)
'                     SC110 DTE shield (clear)
'                     WindMaster Pro shield (clear)

'B&B 4WSD9TB TD B(+)  WindMaster Pro RxB(+) (blue)
'B&B 4WSD9TB TD A(-)  WindMaster Pro RxA(-) (green)
'B&B 4WSD9TB RD B(+)  WindMaster Pro TxB(+) (yellow)
'B&B 4WSD9TB RD A(-)  WindMaster Pro TxA(+) (white)
'B&B 4WSD9TB GND      WindMaster Pro Comms Ref (black)
'                     WindMaster Pro RS-422 Serial Mode (red on signal cable)

'Dip Switches         RS-422; Echo; 4-wire; 4-wire

'*** Wiring Power for both RS-232 and RS-422 ***

'POWER IN
'12V     datalogger (red)
'G       datalogger (black)

'EXTERNAL POWER SUPPLY
```

```
'POS     WindMaster Pro power (red)
'        datalogger (red)
'NEG     WindMaster Pro power reference  (black)
'        datalogger (black)

PipeLineMode

Const SCAN_INTERVAL = 30    'Scan interval in mSec.
Const SONIC_AZIMUTH = 180    'Unique value, azimuth of the -x axis (opposite of the
North marker).
Const OUTPUT_INTERVAL = 30   'Output interval in minutes.

Const SERIAL_BUFF_SIZE = 48*(Ceiling (32*SCAN_INTERVAL/1000))+48+1 'Minimum serial
buffer size is 48 bytes/record * (Ceiling (32 records/second * SCAN_INTERVAL / 1000)) +
48 + 1 bytes.

'WindMaster Pro data.
Public windC(5)
Alias windC(1) = UxC
Alias windC(2) = UyC
Alias windC(3) = UzC
Alias windC(4) = TsC
Alias windC(5) = diag_3dC
Units windC = m/s
Units TsC = C
Units diag_3dC = unitless

Public windB(5)
Alias windB(1) = UxB
Alias windB(2) = UyB
Alias windB(3) = UzB
Alias windB(4) = TsB
Alias windB(5) = diag_3dB
Units windB = m/s
Units TsB = C
Units diag_3dB = unitless

Public windA(5)
Alias windA(1) = UxA
Alias windA(2) = UyA
Alias windA(3) = UzA
Alias windA(4) = TsA
Alias windA(5) = diag_3dA
Units windA = m/s
Units TsA = C
Units diag_3dA = unitless

Public wind_outC(8) 'Wind vector output.
'Alias wind_out(1) = wnd_spd - in compass coordinate system, same As CSAT3.
'Alias wind_out(2) = rslt_wnd_spd - in compass coordinate system, same As CSAT3.
Alias wind_outC(3) = wnd_dir_compassC
'Alias wind_out(4) = std_wnd_dir - in compass coordinate system, same As CSAT3.
Alias wind_outC(5) = wnd_spdC
Alias wind_outC(6) = rslt_wnd_spdC
Alias wind_outC(7) = wnd_dir_sonicC
Alias wind_outC(8) = std_wnd_dirC
Units wnd_dir_compassC = degrees
Units wnd_spdC = m/s
Units rslt_wnd_spdC = m/s
Units wnd_dir_sonicC = degrees
Units std_wnd_dirC= degrees

Public wind_outB(8) 'Wind vector output.
'Alias wind_out(1) = wnd_spd - in compass coordinate system, same As CSAT3.
'Alias wind_out(2) = rslt_wnd_spd - in compass coordinate system, same As CSAT3.
Alias wind_outB(3) = wnd_dir_compassB
'Alias wind_out(4) = std_wnd_dir - in compass coordinate system, same As CSAT3.
Alias wind_outB(5) = wnd_spdB
Alias wind_outB(6) = rslt_wnd_spdB
Alias wind_outB(7) = wnd_dir_sonicB
```

```
Alias wind_outB(8) = std_wnd_dirB
Units wnd_dir_compassB = degrees
Units wnd_spdB = m/s
Units rslt_wnd_spdB = m/s
Units wnd_dir_sonicB = degrees
Units std_wnd_dirB= degrees

Public wind_outA(8) 'Wind vector output.
'Alias wind_out(1) = wnd_spd - in compass coordinate system, same As CSAT3.
'Alias wind_out(2) = rslt_wnd_spd - in compass coordinate system, same As CSAT3.
Alias wind_outA(3) = wnd_dir_compassA
'Alias wind_out(4) = std_wnd_dir - in compass coordinate system, same As CSAT3.
Alias wind_outA(5) = wnd_spdA
Alias wind_outA(6) = rslt_wnd_spdA
Alias wind_outA(7) = wnd_dir_sonicA
Alias wind_outA(8) = std_wnd_dirA
Units wnd_dir_compassA = degrees
Units wnd_spdA = m/s
Units rslt_wnd_spdA = m/s
Units wnd_dir_sonicA = degrees
Units std_wnd_dirA= degrees

Dim in_bytes_str As String * 45
Dim nmbr_bytes_rtrnd
Dim gill_checksum_flg As Boolean
Dim gill_disable_flg As Boolean
Dim n
Dim wind_eastC  'East wind in compass coordinate system.
Dim wind_northC 'North wind in compass coordinate system.
Dim wind_eastB  'East wind in compass coordinate system.
Dim wind_northB 'North wind in compass coordinate system.
Dim wind_eastA  'East wind in compass coordinate system.
Dim wind_northA 'North wind in compass coordinate system.

'Time series data.
DataTable (ts_dataC,TRUE,-1)
  DataInterval (0,SCAN_INTERVAL,mSec,100)
  TableFile ("USB:D:\WMData\WMC\WMC",8,-1,0,1,Hr,0,0) 'New table each hour
  Sample (5,UxC,IEEE4)
  Sample (1,wnd_dir_compassC,IEEE4)
  Sample (4,wnd_spdC,IEEE4)
EndTable

DataTable (ts_dataB,TRUE,-1)
  DataInterval (0,SCAN_INTERVAL,mSec,100)
  TableFile ("USB:D:\WMData\WMB\WMB",8,-1,0,1,Hr,0,0) 'New table each hour
  Sample (5,UxB,IEEE4)
  Sample (1,wnd_dir_compassB,IEEE4)
  Sample (4,wnd_spdB,IEEE4)
EndTable

DataTable (ts_dataA,TRUE,-1)
  DataInterval (0,SCAN_INTERVAL,mSec,100)
  TableFile ("USB:D:\WMData\WMA\WMA",8,-1,0,1,Hr,0,0) 'New table each hour
  Sample (5,UxA,IEEE4)
  Sample (1,wnd_dir_compassA,IEEE4)
  Sample (4,wnd_spdA,IEEE4)
EndTable

'Online statistics.
DataTable (statsC,TRUE,-1)
  DataInterval (0,SCAN_INTERVAL,mSec,100)
  TableFile ("USB:D:\WMData\WMC\WMCStats",8,-1,0,1,Hr,0,0) 'New table each hour
  Sample (1,wnd_dir_compassC,IEEE4)
  Sample (4,wnd_spdC,IEEE4)
  Average (4,UxC,IEEE4,gill_disable_flg)
  Totalize (1,n,IEEE4,gill_disable_flg)
  FieldNames ("n_Tot") 'Number of samples in statistics.
  Totalize (1,n,IEEE4,diag_3dC<>1)
  FieldNames ("diag_1_Tot")
```

```
  Totalize (1,n,IEEE4,diag_3dC<>2)
  FieldNames ("diag_2_Tot")
  Totalize (1,n,IEEE4,diag_3dC<>3)
  FieldNames ("diag_3_Tot")
  Totalize (1,n,IEEE4,diag_3dC<>4)
  FieldNames ("diag_4_Tot")
  Totalize (1,n,IEEE4,diag_3dC<>5)
  FieldNames ("diag_5_Tot")
  Totalize (1,n,IEEE4,diag_3dC<>6)
  FieldNames ("diag_6_Tot")
  Totalize (1,n,IEEE4,diag_3dC<>7)
  FieldNames ("diag_7_Tot")
  Totalize (1,n,IEEE4,diag_3dC<>NaN)
  FieldNames ("n_nnd_Tot") 'No New Data (nnd).
  Totalize (1,n,IEEE4,nmbr_bytes_rtrnd<>0 IMP gill_checksum_flg)
  FieldNames ("n_gill_checksum_err_Tot") 'Checksum errors.
EndTable

DataTable (statsB,TRUE,-1)
  DataInterval (0,SCAN_INTERVAL,mSec,100)
  TableFile ("USB:D:\WMData\WMB\WMBStats",8,-1,0,1,Hr,0,0) 'New table each hour
  Sample (1,wnd_dir_compassB,IEEE4)
  Sample (4,wnd_spdB,IEEE4)
  Average (4,UxB,IEEE4,gill_disable_flg)
  Totalize (1,n,IEEE4,gill_disable_flg)
  FieldNames ("n_Tot") 'Number of samples in statistics.
  Totalize (1,n,IEEE4,diag_3dB<>1)
  FieldNames ("diag_1_Tot")
  Totalize (1,n,IEEE4,diag_3dB<>2)
  FieldNames ("diag_2_Tot")
  Totalize (1,n,IEEE4,diag_3dB<>3)
  FieldNames ("diag_3_Tot")
  Totalize (1,n,IEEE4,diag_3dB<>4)
  FieldNames ("diag_4_Tot")
  Totalize (1,n,IEEE4,diag_3dB<>5)
  FieldNames ("diag_5_Tot")
  Totalize (1,n,IEEE4,diag_3dB<>6)
  FieldNames ("diag_6_Tot")
  Totalize (1,n,IEEE4,diag_3dB<>7)
  FieldNames ("diag_7_Tot")
  Totalize (1,n,IEEE4,diag_3dB<>NaN)
  FieldNames ("n_nnd_Tot") 'No New Data (nnd).
  Totalize (1,n,IEEE4,nmbr_bytes_rtrnd<>0 IMP gill_checksum_flg)
  FieldNames ("n_gill_checksum_err_Tot") 'Checksum errors.
EndTable

DataTable (statsA,TRUE,-1)
  DataInterval (0,SCAN_INTERVAL,mSec,100)
  TableFile ("USB:D:\WMData\WMA\WMAStats",8,-1,0,1,Hr,0,0) 'New table each hour
  Sample (1,wnd_dir_compassA,IEEE4)
  Sample (4,wnd_spdA,IEEE4)
  Average (4,UxA,IEEE4,gill_disable_flg)
  Totalize (1,n,IEEE4,gill_disable_flg)
  FieldNames ("n_Tot") 'Number of samples in statistics.
  Totalize (1,n,IEEE4,diag_3dA<>1)
  FieldNames ("diag_1_Tot")
  Totalize (1,n,IEEE4,diag_3dA<>2)
  FieldNames ("diag_2_Tot")
  Totalize (1,n,IEEE4,diag_3dA<>3)
  FieldNames ("diag_3_Tot")
  Totalize (1,n,IEEE4,diag_3dA<>4)
  FieldNames ("diag_4_Tot")
  Totalize (1,n,IEEE4,diag_3dA<>5)
  FieldNames ("diag_5_Tot")
  Totalize (1,n,IEEE4,diag_3dA<>6)
  FieldNames ("diag_6_Tot")
  Totalize (1,n,IEEE4,diag_3dA<>7)
  FieldNames ("diag_7_Tot")
  Totalize (1,n,IEEE4,diag_3dA<>NaN)
  FieldNames ("n_nnd_Tot") 'No New Data (nnd).
```

```
  Totalize (1,n,IEEE4,nmbr_bytes_rtrnd<>0 IMP gill_checksum_flg)
  FieldNames ("n_gill_checksum_err_Tot") 'Checksum errors.
EndTable

DataTable (windC,TRUE,-1)
  DataInterval (0,SCAN_INTERVAL,mSec,100)
  TableFile ("USB:D:\WMData\WMC\WMCWV",8,-1,0,1,Hr,0,0) 'New table each hour
  WindVector (1,wind_eastC,wind_northC,IEEE4,gill_disable_flg,0,1,2)
  WindVector (1,UyC,UxC,IEEE4,gill_disable_flg,0,1,2)
EndTable

DataTable (windB,TRUE,-1)
  DataInterval (0,SCAN_INTERVAL,mSec,100)
  TableFile ("USB:D:\WMData\WMB\WMBWV",8,-1,0,1,Hr,0,0) 'New table each hour
  WindVector (1,wind_eastB,wind_northB,IEEE4,gill_disable_flg,0,1,2)
  WindVector (1,UyB,UxB,IEEE4,gill_disable_flg,0,1,2)
EndTable

DataTable (windA,TRUE,-1)
  DataInterval (0,SCAN_INTERVAL,mSec,100)
  TableFile ("USB:D:\WMData\WMA\WMAWV",8,-1,0,1,Hr,0,0) 'New table each hour
  WindVector (1,wind_eastA,wind_northA,IEEE4,gill_disable_flg,0,1,2)
  WindVector (1,UyA,UxA,IEEE4,gill_disable_flg,0,1,2)
EndTable

BeginProg
  '38400 bps, no parity, 8 data bits, 1 start bit, 1 stop bit, no flow control
  n = 1
  Scan (SCAN_INTERVAL,mSec,3,0)
    SerialOpen (ComC5,38400,19,0,SERIAL_BUFF_SIZE)
    SerialInRecord (ComC5,in_bytes_str,&h02,0,&h0D0A,nmbr_bytes_rtrnd,01) 'Capture all
the data between the WindMaster Pro start of text (&h02) and the <CR><LF> (&h0D0A).
    SplitStr (UxC,in_bytes_str,"",5,0) 'Split the string and convert to floats.
    gill_checksum_flg = ( (HexToDec (Right (in_bytes_str,2))) EQV (CheckSum
(in_bytes_str,9,Len (in_bytes_str)-3)) )
    gill_disable_flg = (NOT (gill_checksum_flg) OR (nmbr_bytes_rtrnd=0) OR
(diag_3dC<>0))
    'Copy and convert WindMaster Pro for compass wind vector computation.
    wind_eastC = -1*UyC
    wind_northC = UxC
    CallTable ts_dataC
    CallTable windC
    If ( windC.Output(1,1) ) Then
      GetRecord (wind_outC(1),windC,1)
      'Compass wind direction will be between 0 and 360 degrees.
      wnd_dir_compassC = (wnd_dir_compassC+SONIC_AZIMUTH) MOD 360
      'Gill wind direction will be between 0 to 180 degrees and 0 to -180 degrees.
      If ( wnd_dir_sonicC ) > 180 Then ( wnd_dir_sonicC = wnd_dir_sonicC-360 )
    EndIf
    CallTable statsC
    SerialOpen (ComC3,38400,19,0,SERIAL_BUFF_SIZE)
    SerialInRecord (ComC3,in_bytes_str,&h02,0,&h0D0A,nmbr_bytes_rtrnd,01) 'Capture all
the data between the WindMaster Pro start of text (&h02) and the <CR><LF> (&h0D0A).
    SplitStr (UxB,in_bytes_str,"",5,0) 'Split the string and convert to floats.
    gill_checksum_flg = ( (HexToDec (Right (in_bytes_str,2))) EQV (CheckSum
(in_bytes_str,9,Len (in_bytes_str)-3)) )
    gill_disable_flg = (NOT (gill_checksum_flg) OR (nmbr_bytes_rtrnd=0) OR
(diag_3dB<>0))
    'Copy and convert WindMaster Pro for compass wind vector computation.
    wind_eastB = -1*UyB
    wind_northB = UxB
    CallTable ts_dataB
    CallTable windB
    If ( windB.Output(1,1) ) Then
      GetRecord (wind_outB(1),windB,1)
      'Compass wind direction will be between 0 and 360 degrees.
      wnd_dir_compassB = (wnd_dir_compassB+SONIC_AZIMUTH) MOD 360
      'Gill wind direction will be between 0 to 180 degrees and 0 to -180 degrees.
      If ( wnd_dir_sonicB ) > 180 Then ( wnd_dir_sonicB = wnd_dir_sonicB-360 )
    EndIf
```

```
    CallTable statsB
    SerialOpen (ComC1,38400,19,0,SERIAL_BUFF_SIZE)
    SerialInRecord (ComC1,in_bytes_str,&h02,0,&h0D0A,nmbr_bytes_rtrnd,01) 'Capture all
the data between the WindMaster Pro start of text (&h02) and the <CR><LF> (&h0D0A).
    SplitStr (UxA,in_bytes_str,"",5,0) 'Split the string and convert to floats.
    gill_checksum_flg = ( (HexToDec (Right (in_bytes_str,2))) EQV (CheckSum
(in_bytes_str,9,Len (in_bytes_str)-3)) )
    gill_disable_flg = (NOT (gill_checksum_flg) OR (nmbr_bytes_rtrnd=0) OR
(diag_3dA<>0))
    'Copy and convert WindMaster Pro for compass wind vector computation.
    wind_eastA = -1*UyA
    wind_northA = UxA
    CallTable ts_dataA
    CallTable windA
    If ( windA.Output(1,1) ) Then
      GetRecord (wind_outA(1),windA,1)
      'Compass wind direction will be between 0 and 360 degrees.
      wnd_dir_compassA = (wnd_dir_compassA+SONIC_AZIMUTH) MOD 360
      'Gill wind direction will be between 0 to 180 degrees and 0 to -180 degrees.
      If ( wnd_dir_sonicA ) > 180 Then ( wnd_dir_sonicA = wnd_dir_sonicA-360 )
    EndIf
    CallTable statsA
  NextScan
EndProg
extScan
EndProg
```

Table 7.6: Example python code used for atmospheric observation analysis, developed on the Spyder Integrated Development Environment (IDE). The code includes sections to automatically retrieve, filter, analyze and plot data.

```
# -*- coding: utf-8 -*-
"""
Created on Sat Oct 22 14:03:42 2022
@author: Justin Ruedebusch
"""

import matplotlib.pyplot as plt
from windrose import WindroseAxes, WindAxes
import matplotlib.cm as cm
import numpy as np
import pandas as pd
import glob
# import seaborn as sns

'''set the date for data retrieval, ** is used as a wildcard, ex. if 2022-03 is used
all date from March will be retrieved'''
year = '2022'
month = '03'
day = '08'
hour = '15**'

'''pulls data from UMD Wind Turbine shared google drive'''
'''MetSens A Import'''
def msa_data(year,month,day,hour): #defines a function
    path = r'G:\Shared drives\UMD Wind Turbine\Met
Data\CSLogger\MSA_'+year+'_'+month+'_'+day+'_'+hour+'.csv' #path at which data is
stored, may need to be switched to .dat
    allFiles = sorted(glob.glob(path)) #ensures data is imported from old->new
    msa = pd.DataFrame() #defines dataframe to store data in
    list_ = [] #creates empty list
    for file_ in allFiles: #loop to import all files with
      msa = pd.read_csv(file_,delimiter=',', header=1, index_col=1) #specifies file
format, comma seperated header in line 1, indexed from column 1
      msa['TMSTAMP'] = pd.to_datetime(msa['TMSTAMP'], format='%Y-%m-%d %H:%M:%S.%f')
#Formats timestamp of file
      list_.append(msa) #combines all files together
      for x in msa.index: #loop to drop bad values
```

```
        if msa.loc[x, 'WS_msA'] == -7999: #finds all windspeed values that were
invalid, often associated with other bad values
          msa.drop(x, inplace = True) #prevents creation of a seperate dataframe
    msa = pd.concat(list_) #creates single lsit of data frames from files
    return msa #returns single dataframe
'''MetSens B Import'''
def msb_data(year,month,day,hour):
    path = r'G:/Shared drives/UMD Wind Turbine/Met
Data/CSLogger/MSB_'+year+'_'+month+'_'+day+'_'+hour+'.csv'
    allFiles = sorted(glob.glob(path))
    msb = pd.DataFrame()
    list_ = []
    for file_ in allFiles:
      msb = pd.read_csv(file_,delimiter=',', header=1, index_col=1)
      msb['TMSTAMP'] = pd.to_datetime(msb['TMSTAMP'], format='%Y-%m-%d %H:%M:%S.%f')
      list_.append(msb)
      for x in msb.index:
        if msb.loc[x, 'WS_msB'] == -7999:
          msb.drop(x, inplace = True)
    msb = pd.concat(list_)
    return msb
'''MetSens C Import'''
def msc_data(year,month,day,hour):
    path = r'G:/Shared drives/UMD Wind Turbine/Met
Data/CSLogger/MSC_'+year+'_'+month+'_'+day+'_'+hour+'.csv'
    allFiles = sorted(glob.glob(path))
    msc = pd.DataFrame()
    list_ = []
    for file_ in allFiles:
      msc = pd.read_csv(file_,delimiter=',', header=1, index_col=1)
      msc['TMSTAMP'] = pd.to_datetime(msc['TMSTAMP'], format='%Y-%m-%d %H:%M:%S.%f')
      list_.append(msc)
      for x in msc.index:
        if msc.loc[x, 'WS_msC'] == -7999:
          msc.drop(x, inplace = True)
    msc = pd.concat(list_)
    return msc
'''Solar Pyranometer Import'''
def solar_data(year,month,day,hour):
    path = r'G:/Shared drives/UMD Wind Turbine/Met
Data/CSLogger/Solar_'+year+'_'+month+'_'+day+'_'+hour+'.csv'
    allFiles = sorted(glob.glob(path))
    solar = pd.DataFrame()
    list_ = []
    for file_ in allFiles:
     solar = pd.read_csv(file_,delimiter=',', header=1, index_col=1)
     solar['TMSTAMP'] = pd.to_datetime(solar['TMSTAMP'], format='%Y-%m-%d %H:%M:%S.%f')
     list_.append(solar)
     for x in solar.index:
       if solar.loc[x, 'Raw_mV_Avg'] == -7999:
         solar.drop(x, inplace = True)
    solar = pd.concat(list_)
    return solar
'''WindMaster A Import'''
def wma_data(year,month,day,hour):
    path = r'G:/Shared drives/UMD Wind Turbine/Met
Data/WMLogger/WMA_'+year+'_'+month+'_'+day+'_'+hour+'.csv'
    allFiles = sorted(glob.glob(path))
    wma = pd.DataFrame()
    list_ = []
    for file_ in allFiles:
      wma = pd.read_csv(file_,delimiter=',', header=1, index_col=1)
      wma['TMSTAMP'] = pd.to_datetime(wma['TMSTAMP'], format='%Y-%m-%d %H:%M:%S.%f')
      list_.append(wma)
      for x in wma.index:
        if wma.loc[x, 'TsA'] == -7999:
          wma.drop(x, inplace = True)
    wma = pd.concat(list_)
    return wma
'''WindMaster B Import'''
```

```
def wmb_data(year,month,day,hour):
    path = r'G:/Shared drives/UMD Wind Turbine/Met
Data/WMLogger/WMB_'+year+'_'+month+'_'+day+'_'+hour+'.csv'
    allFiles = sorted(glob.glob(path))
    wmb = pd.DataFrame()
    list_ = []
    for file_ in allFiles:
      wmb = pd.read_csv(file_,delimiter=',', header=1, index_col=1)
      wmb['TMSTAMP'] = pd.to_datetime(wmb['TMSTAMP'], format='%Y-%m-%d %H:%M:%S.%f')
      list_.append(wmb)
      for x in wmb.index:
        if wmb.loc[x, 'TsB'] == -7999:
          wmb.drop(x, inplace = True)
    wmb = pd.concat(list_)
    return wmb

'''Windmaster C Import'''
def wmc_data(year,month,day,hour):
    path = r'G:/Shared drives/UMD Wind Turbine/Met
Data/WMLogger/WMC_'+year+'_'+month+'_'+day+'_'+hour+'.csv'
    allFiles = sorted(glob.glob(path))
    wmc = pd.DataFrame()
    list_ = []
    for file_ in allFiles:
      wmc = pd.read_csv(file_,delimiter=',', header=1, index_col=1)
      wmc['TMSTAMP'] = pd.to_datetime(wmc['TMSTAMP'], format='%Y-%m-%d %H:%M:%S.%f')
      list_.append(wmc)
      for x in wmc.index:
        if wmc.loc[x, 'TsC'] == -7999:
          wmc.drop(x, inplace = True)
    wmc = pd.concat(list_)
    return wmc
# '''CSAT3BH Import'''

'''Defines and creates datafram from Import to use locally'''
msa = msa_data(year,month,day,hour) #MetSens A
msb = msb_data(year,month,day,hour) #MetSens B
msc = msc_data(year,month,day,hour) #MetSens C
wma = wma_data(year,month,day,hour) #Windmaster A
wmb = wmb_data(year,month,day,hour) #Windmaster B
wmc = wmc_data(year,month,day,hour) #Windmaster C
# ds = ds_data(year,month,day,hour) #CSAT
solar = solar_data(year,month,day,hour) #Solar Pyranometer

figureIndex = 1 #creates index for multiple figures

'''Data Cleaning'''
msa.rename(columns={'BP_mbarA': 'bp', 'RHA': 'rh','AirT_CA': 'AT', 'DP_CA':
'dp','WindDirA': 'wd', 'WS_msA': 'ws'}, inplace=True) #Renames dataframe columns to
work with easier
msb.rename(columns={'BP_mbarB': 'bp', 'RHB': 'rh','AirT_CB': 'AT', 'DP_CB':
'dp','WindDirB': 'wd', 'WS_msB': 'ws'}, inplace=True) #Renames dataframe columns to
work with easier
msc.rename(columns={'BP_mbarC': 'bp', 'RHC': 'rh','AirT_CC': 'AT', 'DP_CC':
'dp','WindDirC': 'wd', 'WS_msC': 'ws'}, inplace=True) #Renames dataframe columns to
work with easier
wma.rename(columns={'UxA': 'u', 'UyA': 'v','UzA': 'w', 'TsA': 'AT'}, inplace=True)
#Renames dataframe columns to work with easier
wmb.rename(columns={'UxB': 'u', 'UyB': 'v','UzB': 'w', 'TsB': 'AT'}, inplace=True)
#Renames dataframe columns to work with easier
wmc.rename(columns={'UxC': 'u', 'UyC': 'v','UzC': 'w', 'TsC': 'AT'}, inplace=True)
#Renames dataframe columns to work with easier
'''MetSens A'''
drop_MSA = ['MetSENS_StatusA','CommResultA'] #Drops columns not used
msa.drop(drop_MSA, inplace = True, axis = 1) #Removes from the initial dataframe
for x in msa.index: #loop to clean data
  if msa.loc[x, 'bp'] == -7999: #drops bad atmospheric data
    msa.drop(x, inplace = True) #drops inplace
'''MetSens B'''
drop_MSB = ['MetSENS_StatusB','CommResultB']
```

```
msb.drop(drop_MSB, inplace = True, axis = 1)
for x in msb.index:
  if msb.loc[x, 'bp'] == -7999:
    msb.drop(x, inplace = True)
'''MetSens C'''
drop_MSC = ['MetSENS_StatusC','CommResultC']
msc.drop(drop_MSC, inplace = True, axis = 1)
for x in msc.index:
  if msc.loc[x, 'bp'] == -7999:
    msc.drop(x, inplace = True)
'''Windmaster A'''
for x in wma.index:
  if wma.loc[x, 'AT'] == 999.99:
    wma.drop(x, inplace = True)
'''Windmaster B'''
for x in wmb.index:
  if wmb.loc[x, 'AT'] == 999.99:
    wmb.drop(x, inplace = True)
'''Windmaster C'''
for x in wmc.index:
  if wmc.loc[x, 'AT'] == 999.99:
    wmc.drop(x, inplace = True)

'''Name Convention'''
'''
sensor_measurement_operation

Sensors
msa
msb
msc
wma
wmb
wmc
ds

Windspeed: ws
Wind direction: wd
Air temp: AT
Barometric Pressure: bp
Relative Humidity: rh
dew point: dp

Operation
avgerage: avg
standard deviation: std
turbulence intensity: ti
'''
'''Plotting Legend'''
'''Can comment a section out to easily change style'''

'''lines'''
msa_ls = ':' #
msb_ls = '--'#
msc_ls = '-.' #
wma_ls =  ':'#
wmb_ls = '--'#
wmc_ls = '-.'#
ds_ls =  '-'#

# msa_ls = 'None' #
# msb_ls = 'None'#
# msc_ls = 'None'#
# # wma_ls = 'None'#
# # wmb_ls = 'None'#
# # wmc_ls = 'None'#
# ds_ls = 'None' #

'''markers'''
# msa_marker = 'o' #
```

```
# msb_marker = '*'#
# msc_marker = '+'#
# wma_marker = '^' #
# wmb_marker = 'D'#
# wmc_marker = 'x' #
# ds_marker = ','

msa_marker = 'None' #
msb_marker = 'None' #
msc_marker = 'None' #
wma_marker = 'None' #
wmb_marker = 'None' #
wmc_marker = 'None' #
ds_marker = 'None' #

'''color'''
msa_color = 'salmon'
msb_color = 'deepskyblue'
msc_color = 'mediumseagreen'
wma_color = 'maroon'
wmb_color = 'royalblue'
wmc_color = 'forestgreen'
ds_color = 'gold'

'''heatmap colors'''
a_cmap = cm.Reds
b_cmap = cm.Blues
c_cmap = cm.Greens
ds_cmap= cm.YlOrBr

plt.rcParams['font.sans-serif'] = "Times New Roman"
plt.rcParams['font.family'] = "sans-serif"

'''MSA Analysis'''
msa['bp'] += 47.76241 #correction to barometric pressure based on height on tower
msa['wd'] = np.mod(msa.wd + (360 - msa.Mag_HeadA),360) #corrects wind direction based
on magnetic heading
msa['wd_avg'] = msa.wd.rolling(600, min_periods=600).mean() #direction 10 min rolling
average
msa['ws_avg'] = msa.ws.rolling(600, min_periods=600).mean() #speed 10m rolling average
msa['ws_std_avg'] = msa.ws.rolling(600, min_periods=600).std() #standard deviation 10m
rolling average
msa['ti'] = (msa['ws_std_avg']/msa['ws_avg'] *100) #turbelence intensity of 10m rolling
average
msa['AT_avg'] = msa.AT.rolling(600, min_periods=600).mean() #air temperature 10m
rolling average
# msa.set_index("TMSTAMP", inplace = True) #index timestamp to resample at in intervals
# # msa_resample = pd.DataFrame(index = pd.date_range('2022-03-08','2022-03-08',
freq='min'))
# msa_resample = msa.resample('min', closed ='left').mean()
# msa_resample_std = msa.resample('min', closed ='right').std()

'''MSB Analysis'''
msb['bp'] += 48.46646
msb['wd'] = np.mod(msb.wd + (360 - msb.Mag_HeadB),360)
msb['wd_avg'] = msb.wd.rolling(600, min_periods=600).mean() # direction 10 min average
msb['ws_avg'] = msb.ws.rolling(600, min_periods=600).mean() #speed 10m average
msb['ws_std_avg'] = msb.ws.rolling(600, min_periods=600).std()
msb['ti'] = (msb['ws_std_avg']/msb['ws_avg'] *100)#.rolling(600,
min_periods=600).mean()
msb['AT_avg'] = msb.AT.rolling(600, min_periods=600).mean()
msb['tke'] = (msb['ws_std_avg']*msb['ws_std_avg'])/2
# msb.set_index("TMSTAMP", inplace = True)
# # msb_resample = pd.DataFrame(index = pd.date_range('2022-03-08','2022-03-08',
freq='min'))
# msb_resample = msb.resample('min', closed ='left').mean()
# msb_resample_std = msb.resample('min', closed ='right').std()

'''MSC Analysis'''
msc['bp'] += 49.1701
```

```
msc['wd'] = np.mod(msc.wd + msc.Mag_HeadC,360)
msc['wd_avg'] = msc.wd.rolling(600, min_periods=600).mean() # direction 10 min average
msc['ws_avg'] = msc.ws.rolling(600, min_periods=600).mean() #speed 10m average
msc['ws_std'] = msc.ws.rolling(600, min_periods=600).std()
msc['ws_std_avg'] = msc.ws.rolling(600, min_periods=600).std()
msc['ti'] = (msc['ws_std_avg']/msc['ws_avg']*100)#.rolling(600, min_periods=600).mean()
# msc['ti'] = msc.ti.rolling(600, min_periods=600).mean
msc['AT_avg'] = msc.AT.rolling(600, min_periods=600).mean()
msc['tke'] = (msc['ws_std_avg']*msc['ws_std_avg'])/2
# msc['u'] = (-msc['ws']*np.sin(msc['wd']*np.pi/180))
# msc['v'] = (-msc['ws']*np.cos(msc['wd']*np.pi/180))
# msc['ws_calc'] = np.sqrt((msc['u']*msc['u']) + (msc['v']*msc['v']))
# msc['wd_calc'] =np.mod(180+np.rad2deg(np.arctan2(msc['u'],msc['v'])),360)

# msc.set_index("TMSTAMP", inplace = True)
# # msc_resample = pd.DataFrame(index = pd.date_range('2022-03-08','2022-03-08',
freq='min'))
# msc_resample = msc.resample('min', closed ='left').mean()
# msc_resample_std = msc.resample('min', closed ='right').std()

'''WMA Analysis'''
wma['ws'] = np.sqrt((wma['u']*wma['u']) + (wma['v']*wma['v'])) #horizontal wind speed
wma['wd'] =np.mod(5+np.rad2deg(np.arctan2(wma['v'],-wma['u'])),360) #wind direction
with 5 degree correction
wma['x'] = np.round(-wma['ws'] * np.sin(wma['wd'] * np.pi / 180), 8) #calculate
corrected x component, round to 8 digits
wma['y'] = np.round(-wma['ws'] * np.cos(wma['wd'] * np.pi / 180), 8)  #calculate
corrected y component, round to 8 digits
wma['ws'] = np.sqrt((wma['x']*wma['x']) + (wma['y']*wma['y'])) #calculate wind speed
with corrected direction
wma['wd_avg'] = wma.wd.rolling(19200, min_periods=19200).mean()# direction 10 min
rolling average
wma['ws_avg'] = wma.ws.rolling(19200, min_periods=19200).mean() #speed 10m rolling
average
wma['ws_std_avg'] = wma.ws.rolling(19200, min_periods=19200).std() #standard deviation
of windspeed
wma['ti'] = wma['ws_std_avg']/wma['ws_avg'] *100 #tubulence intensity from rolling
average
wma['AT_avg'] = wma.AT.rolling(19200, min_periods=19200).mean() # 10m rolling average
sonic temp
# wma.set_index("TMSTAMP", inplace = True)
# # wma_resample = pd.DataFrame(index = pd.date_range('2022-03-08','2022-03-08',
freq='min'))
# wma_resample = wma.resample('min', closed ='left').mean()
# wma_resample_std = wma.resample('min', closed ='right').std()

'''WMB Analysis'''
wmb['ws'] = np.sqrt((wmb['u']*wmb['u']) + (wmb['v']*wmb['v']))
wmb['wd'] =np.mod(-35+np.rad2deg(np.arctan2(wmb['v'],-wmb['u'])),360) # wind direction
with 30 degree axis correction and -5 degree misalignment correction
wmb['x'] = np.round(-wmb['ws'] * np.sin(wmb['wd'] * np.pi / 180), 8)
wmb['y'] = np.round(-wmb['ws'] * np.cos(wmb['wd'] * np.pi / 180), 8)
wmb['ws'] = np.sqrt((wmb['x']*wmb['x']) + (wmb['y']*wmb['y']))
wmb['wd_avg'] = wmb.wd.rolling(19200, min_periods=19200).mean()
wmb['ws_avg'] = wmb.ws.rolling(19200, min_periods=19200).mean()
wmb['ws_std_avg'] = wmb.ws.rolling(19200, min_periods=19200).std()
wmb['ti'] = wmb['ws_std_avg']/wmb['ws_avg'] *100
wmb['AT_avg'] = wmb.AT.rolling(19200, min_periods=19200).mean()
# wmb.set_index("TMSTAMP", inplace = True)
# # wmb_resample = pd.DataFrame(index = pd.date_range('2022-03-08','2022-03-08',
freq='min'))
# wmb_resample = wmb.resample('min', closed ='left').mean()
# wmb_resample_std = wmb.resample('min', closed ='right').std()

'''WMC Analysis'''
wmc['ws'] = np.sqrt((wmc['u']*wmc['u']) + (wmc['v']*wmc['v']))
wmc['wd'] =np.mod(-35+np.rad2deg(np.arctan2(wmc['v'],-wmc['u'])),360) # wind direction
with 30 degree axis correction and -5 degree misalignment correction
wmc['x'] = np.round(-wmc['ws'] * np.sin(wmc['wd'] * np.pi / 180), 8)
wmc['y'] = np.round(-wmc['ws'] * np.cos(wmc['wd'] * np.pi / 180), 8)
```

```
wmc['ws_adj'] = np.sqrt((wmc['x']*wmc['x']) + (wmc['y']*wmc['y']))
wmc['wd_avg'] = wmc.wd.rolling(19200, min_periods=19200).mean()
wmc['ws_avg'] = wmc.ws.rolling(19200, min_periods=19200).mean()
wmc['ws_std_avg'] = wmc.ws.rolling(19200, min_periods=19200).std()
wmc['ti'] = wmc['ws_std_avg']/wmc['ws_avg'] *100
wmc['AT_avg'] = wmc.AT.rolling(19200, min_periods=19200).mean()
# wmc.set_index("TMSTAMP", inplace = True)
# # wmc_resample = pd.DataFrame(index = pd.date_range('2022-03-08','2022-03-08',
freq='min'))
# wmc_resample = wmc.resample('min', closed ='left').mean()
# wmc_resample_std = wmc.resample('min', closed ='right').std()

'''CSAT Analysis'''
# ds.rename(columns={'Ux': 'u', 'Uy': 'v','Uz': 'w'}, inplace=True) #rename columns to
work with data easier
# ds['ws'] = np.sqrt((ds['u']*ds['u']) + (ds['v']*ds['v'])) #calculate windspeed
# ds['wd'] =np.mod(30+np.rad2deg(np.arctan2(ds['v'],-ds['u'])),360) #calculate
windspeed with axis correction
# ds['x'] = np.round(-ds['ws'] * np.sin(ds['wd'] * np.pi / 180), 8)  #recalculate x
component with correction
# ds['y'] = np.round(-ds['ws'] * np.cos(ds['wd'] * np.pi / 180), 8)  #recalculate y
component with correction
# # ds['ws_adj'] = np.sqrt((ds['x']*ds['x']) + (ds['y']*ds['y'])) #recalculate wind
direction with corrected components to verify x,y
# ds['wd_avg'] = ds.wd.rolling(30000, min_periods=30000).mean()
# ds['ws_avg'] = ds.ws.rolling(30000, min_periods=30000).mean()
# ds['ws_std_avg'] = ds.ws.rolling(30000, min_periods=30000).std()
# # ds.set_index("TMSTAMP", inplace = True)
# # # ds_resample = pd.DataFrame(index = pd.date_range('2022-03-08','2022-03-08',
freq='min'))
# # ds_resample = ds.resample('min', closed ='left').mean()
# # ds_resample_std = ds.resample('min', closed ='right').std()

'''Solar Analysis'''
solar.set_index("TMSTAMP", inplace = True) #setting index as timestamp to resample data
# msa_resample = pd.DataFrame(index = pd.date_range('2022-03-08','2022-03-08',
freq='min')) #to select a timeframe from the imported data
solar_resample = solar.resample('5min', closed ='left').mean() #5min sampled average
# msa_resample_std = msa.resample('min', closed ='right').std() #5min standard
deviation

'''Plotting Wind Direction for All '''
plt.figure(figureIndex)
plt.plot(msa['TMSTAMP'],msa['wd_avg'], linestyle = msa_ls, c = msa_color, marker =
msa_marker, mec = msa_color, mfc = 'None', label = 'MSA (Bottom 20ft)')
plt.plot(msb['TMSTAMP'],msb['wd_avg'], linestyle = msb_ls, c = msb_color, marker =
msb_marker, mec = msb_color, mfc = 'None', label = 'MSB (Middle 40ft)')
plt.plot(msc['TMSTAMP'],msc['wd_avg'], linestyle = msc_ls, c = msc_color, marker =
msc_marker, mec = msc_color, mfc = 'None', label = 'MSC (Top 60ft)')
plt.plot(wma['TMSTAMP'],wma['wd_avg'], linestyle = wma_ls, c = wma_color, marker =
wma_marker, mec = wma_color, mfc = 'None', label = 'WMA (Bottom 20ft)')
plt.plot(wmb['TMSTAMP'],wmb['wd_avg'], linestyle = wmb_ls, c = wmb_color, marker =
wmb_marker, mec = wmb_color, mfc = 'None', label = 'WMB (Middle 40ft)')
plt.plot(wmc['TMSTAMP'],wmc['wd_avg'], linestyle = wmc_ls, c = wmc_color, marker =
wmc_marker, mec = wmc_color, mfc = 'None', label = 'WMC (Top 60ft)')
# plt.plot(ds['TMSTAMP'],ds['wd_avg'], linestyle = ds_ls, c = ds_color, marker =
ds_marker, mec = ds_color, mfc = ds_color, label = 'CSAT (Top 60ft)')
plt.title("Wind Direction")
plt.xlabel("Time")
plt.ylabel("Direction (degrees)")
plt.grid(color = 'black', linestyle = '--', linewidth = 0.5)
plt.legend()
# plt.gcf().autofmt_xdate()
plt.show()
figureIndex = figureIndex + 1

'''Plotting Wind Speed for All '''
plt.figure(figureIndex)
plt.plot(msa['TMSTAMP'],msa['ws_avg'], linestyle = msa_ls, c = msa_color, marker =
msa_marker, mec = msa_color, mfc = 'None', label = 'MSA (Bottom 20ft)')
```

```
plt.plot(msb['TMSTAMP'],msb['ws_avg'], linestyle = msb_ls, c = msb_color, marker =
msb_marker, mec = msb_color, mfc = 'None', label = 'MSB (Middle 40ft)')
plt.plot(msc['TMSTAMP'],msc['ws_avg'], linestyle = msc_ls, c = msc_color, marker =
msc_marker, mec = msc_color, mfc = 'None', label = 'MSC (Top 60ft)')
plt.plot(wma['TMSTAMP'],wma['ws_avg'], linestyle = wma_ls, c = wma_color, marker =
wma_marker, mec = wma_color, mfc = 'None', label = 'WMA (Bottom 20ft)')
plt.plot(wmb['TMSTAMP'],wmb['ws_avg'], linestyle = wmb_ls, c = wmb_color, marker =
wmb_marker, mec = wmb_color, mfc = 'None', label = 'WMB (Middle 40ft)')
plt.plot(wmc['TMSTAMP'],wmc['ws_avg'], linestyle = wmc_ls, c = wmc_color, marker =
wmc_marker, mec = wmc_color, mfc = 'None', label = 'WMC (Top 60ft)')
# plt.plot(ds['TMSTAMP'],ds['ws_avg'], linestyle = ds_ls, c = ds_color, marker =
ds_marker, mec = ds_color, mfc = ds_color, label = 'CSAT (Top 60ft)')
# plt.plot(wma_resample.index,wma_resample['ws_adj'], linestyle = wma_ls, c =
wma_color, marker = wma_marker, mec = wma_color, mfc = 'None', label = 'WMA (Bottom
20ft)')
# plt.plot(wmb_resample.index,wmb_resample['ws_adj'], linestyle = wmb_ls, c =
wmb_color, marker = wmb_marker, mec = wmb_color, mfc = 'None', label = 'WMB (Middle
40ft)')
# plt.plot(wmc_resample.index,wmc_resample['ws_adj'], linestyle = wmc_ls, c =
wmc_color, marker = wmc_marker, mec = wmc_color, mfc = 'None', label = 'WMC (Top
60ft)')
plt.title("Wind Speed")
plt.xlabel("Time")
plt.ylabel("Wind Speed (m/s)")
plt.grid(color = 'black', linestyle = '--', linewidth = 0.5)
plt.legend()
# plt.gcf().autofmt_xdate()
plt.show()
figureIndex = figureIndex + 1

'''Plotting Solar'''
plt.figure(figureIndex)
plt.plot(solar.index,solar['SlrRad_W_Avg'], linestyle = ds_ls, c = ds_color, marker =
ds_marker, mec = ds_color, mfc = 'None', label = 'CS320 (Middle, 40ft)')
plt.title("CS320 Short Wave Radiation 5-minute Averaged Observations")
plt.xlabel("Time")
plt.ylabel("Solar Irradiance (W/m2)")
plt.grid(color = 'black', linestyle = '--', linewidth = 0.5)
plt.legend()
# plt.gcf().autofmt_xdate()
plt.show()
figureIndex = figureIndex + 1

'''Plotting TI for All '''
plt.figure(figureIndex)
plt.plot(msa['TMSTAMP'],msa['ti'], linestyle = msa_ls, c = msa_color, marker =
msa_marker, mec = msa_color, mfc = 'None', label = 'MS A (20ft)')
plt.plot(msb['TMSTAMP'],msb['ti'], linestyle = msb_ls, c = msb_color, marker =
msb_marker, mec = msb_color, mfc = 'None', label = 'MS B (40ft)')
plt.plot(msc['TMSTAMP'],msc['ti'], linestyle = msc_ls, c = msc_color, marker =
msc_marker, mec = msc_color, mfc = 'None', label = 'MS C (60ft)')
plt.plot(wma['TMSTAMP'],wma['ti'], linestyle = wma_ls, c = wma_color, marker =
wma_marker, mec = wma_color, mfc = 'None', label = 'WMA (Bottom 20ft)')
plt.plot(wmb['TMSTAMP'],wmb['ti'], linestyle = wmb_ls, c = wmb_color, marker =
wmb_marker, mec = wmb_color, mfc = 'None', label = 'WMB (Middle 40ft)')
plt.plot(wmc['TMSTAMP'],wmc['ti'], linestyle = wmc_ls, c = wmc_color, marker =
wmc_marker, mec = wmc_color, mfc = 'None', label = 'WMC (Top 60ft)')
plt.title("Turbulence Intensity")
plt.xlabel("Time")
plt.ylabel("Turbulence Intensity (%)")
plt.grid(color = 'black', linestyle = '--', linewidth = 0.5)
plt.legend()
# plt.gcf().autofmt_xdate()
plt.show()
figureIndex = figureIndex + 1

'''TI/WS Heatmap'''
plt.figure(figureIndex)
sns.histplot(data=msc , x='ws', y='ti', stat='percent', bins=30)
plt.title("Heatmap of Top Station (60ft)")
```

```
plt.xlabel("Wind Speed (m/s)")
plt.ylabel("TI (%)")
plt.grid(color = 'black', linestyle = '--', linewidth = 0.5)
plt.show()
figureIndex = figureIndex + 1

'''WS Histogram'''
plt.figure(figureIndex)
sns.histplot(data=msc , x='ws', kde=True, binwidth=0.25)
plt.title("Wind Speed Distrubition at Top Station (60ft)")
plt.xlabel("Wind Speed (m/s)")
plt.ylabel("Count")
plt.grid(color = 'black', linestyle = '--', linewidth = 0.5)
plt.show()
figureIndex = figureIndex + 1

'''TI Histogram'''
plt.figure(figureIndex)
sns.histplot(data=msc , x='ti', binwidth=0.5,stat='percent')
plt.title("Turbulence Intensity at Top Station (60ft)")
plt.xlabel("TI (%)")
plt.ylabel("Count")
plt.grid(color = 'black', linestyle = '--', linewidth = 0.5)
plt.show()
figureIndex = figureIndex + 1

'''Plotting Wind Rose for MetSens A'''
plt.figure(figureIndex, figsize=(12,6))
ax = WindroseAxes.from_ax()
ax.bar(msa['wd'], msa['ws'],blowto=False, normed=True, bins=np.arange(0, 13, 2),
nsector= 32, opening=0.94, cmap=a_cmap, edgecolor='gray', lw=0.1)
ax.set_theta_zero_location("E", offset=0)
ax.set_legend(loc=(-0.08, -0.08), title='Wind Speed (m/s)', prop = 'larger', fontsize =
18)
ax.set_title("MetSens A (Bottom 20ft)",  size=18)
ax.set_thetagrids([0, 22.5, 45, 67.5, 90, 112.5, 135, 157.5, 180, 202.5, 225, 247.5,
270, 292.5, 315, 337.5])
ax.set_xticklabels(['E', 'E-NE', 'NE', 'N-NE', 'N', 'N-NW', 'NW', 'W-NW', 'W', 'W-SW',
'SW', 'S-SW', 'S', 'S-SE', 'SE', 'E-SE'], size = 'larger')
# ax.legend(fontsize=18)
# cx.set_figwidth(6)
# cx.set_figheight(6)
plt.show()
figureIndex = figureIndex + 1

'''Plotting Wind Rose for MetSens B'''
plt.figure(figureIndex,  figsize=(12,6))
bx = WindroseAxes.from_ax()
bx.bar(msb['wd'], msb['ws'], blowto=False, normed=True, bins=np.arange(0, 13, 2),
nsector= 32, opening=0.94, cmap=b_cmap, edgecolor='gray', lw=0.1)
bx.set_theta_zero_location("E", offset=0)
bx.set_legend(loc=(-0.08, -0.08), title='Wind Speed (m/s)', prop = 'larger', fontsize =
18)
bx.set_title("MetSens B (Middle 40ft)",  size=18)
bx.set_thetagrids([0, 22.5, 45, 67.5, 90, 112.5, 135, 157.5, 180, 202.5, 225, 247.5,
270, 292.5, 315, 337.5])
bx.set_xticklabels(['E', 'E-NE', 'NE', 'N-NE', 'N', 'N-NW', 'NW', 'W-NW', 'W', 'W-SW',
'SW', 'S-SW', 'S', 'S-SE', 'SE', 'E-SE'], size = 'larger')
# bx.legend(fontsize=18)
# cx.set_figwidth(6)
# cx.set_figheight(6)
plt.show()
figureIndex = figureIndex + 1

'''Plotting Wind Rose for MetSens C'''
plt.figure(figureIndex,  figsize=(12,6))
cx = WindroseAxes.from_ax()
cx.bar(msc['wd'], msc['ws'],blowto=False, normed=True, bins=np.arange(0, 13, 2),
nsector= 32, opening=0.94, cmap=c_cmap, edgecolor='gray', lw=0.1)
cx.set_theta_zero_location("E", offset=0)
```

```
cx.set_legend(loc=(-0.08, -0.08), title='Wind Speed (m/s)', prop = 'larger', fontsize =
18)
cx.set_title("MetSens C (Top 60ft)",  size=18)
cx.set_thetagrids([0, 22.5, 45, 67.5, 90, 112.5, 135, 157.5, 180, 202.5, 225, 247.5,
270, 292.5, 315, 337.5])
cx.set_xticklabels(['E', 'E-NE', 'NE', 'N-NE', 'N', 'N-NW', 'NW', 'W-NW', 'W', 'W-SW',
'SW', 'S-SW', 'S', 'S-SE', 'SE', 'E-SE'], size = 'larger')
# cx.legend(fontsize=18)
# cx.set_figwidth(6)
# cx.set_figheight(6)
plt.show()
figureIndex = figureIndex + 1
```

Table 7.7: Wiring diagram for both CR1000X datalogger operating all the sensors, outlining the cables from the sensors on the tower, the pass through cables from the tower base to the shed and connections to the dataloggers.

Wire from Tower	**Connector**	**Pass-Through Wire**	**Datalogger**		**Pass-Through**	
			Ethernet-Ethernet			
			SDM:C1,C2,C3		**Identifier**	**Type**
Bundle 1 (A)				A	Black-Red-Black	4-wire
MetSens500		Black-Red-Black		A	Black-Blue-Black	4-wire
SHIELD	6-pin	SHIELD	SDM1:RG	B	Red-Black	4-wire
BLACK	6-pin	BLACK	SDM1:G	B	Red-Blue	4-wire
RED	6-pin	RED	SDM:12V	B	Blue-Black	4-wire
GREEN	6-pin	GREEN	SDM1:TD-	C	Red	4-wire
WHITE	6-pin	WHITE	SDM1:TD+	C	Black	4-wire
Windmaster		Black-Blue-Black		C	Blue	4-wire
SHIELD (3)	Terminal	B/O	G - 24V	C	Brown	Thermocouple
BLACK/BLUE	Terminal	B/O	B - 24V	C	Blue	CPI/Ethernet
RED/YELLOW	Terminal	B/O	W - 24V		Yellow	CPI/Ethernet
RED	6-pin	RED	C2		Orange	Power
WHITE	6-pin	WHITE	C2		Orange	Power
RED	6-pin	BLACK	C1			
GREEN	6-pin	GREEN	C1		**Breakouts**	
					Incoming 24V Ext. Cord	
Bundle 2 (B)					Power	Windmaster A
MetSens500		Red-Black				Windmaster B
SHIELD	6-pin	SHIELD	SDM2:RG			Windmaster C
BLACK	6-pin	BLACK	SDM2:G			CSAT3BH
RED	6-pin	RED	SDM:12V		Ground	Windmaster A
GREEN	6-pin	GREEN	SDM2:TD-			Windmaster B
WHITE	6-pin	WHITE	SDM2:TD+			Windmaster C
Windmaster		Blue-Black				CSAT3BH
SHIELD (3)	Terminal	B/O	G - 24V		Shields	Windmaster A
BLACK/BLUE	Terminal	B/O	B - 24V			Windmaster B
RED/YELLOW	Terminal	B/O	W - 24V			Windmaster C
RED	6-pin	RED	C4			
WHITE	6-pin	WHITE	C4			
RED	6-pin	BLACK	C3			
GREEN	6-pin	GREEN	C3			
CS320		Red-Blue				
SHIELD	6-pin	SHIELD	G			

BLACK	6-pin	BLACK	G
RED	6-pin	RED	12V
WHITE	6-pin	WHITE	C5
BLUE	6-pin	GREEN	G
Bundle 3 (C)			
MetSens500		Red	
SHIELD	6-pin	SHIELD	SDM3:RG
BLACK	6-pin	BLACK	SDM3:G
RED	6-pin	RED	SDM:12V
GREEN	6-pin	GREEN	SDM3:TD-
WHITE	6-pin	WHITE	SDM3:TD+
Windmaster		Blue	
SHIELD (3)	Terminal	B/O	G - 24V
BLACK/BLUE	Terminal	B/O	B - 24V
RED/YELLOW	Terminal	B/O	W - 24V
RED	6-pin	RED	C6
WHITE	6-pin	WHITE	C6
RED	6-pin	BLACK	C5
GREEN	6-pin	GREEN	C5
CSAT3BH			
BLACK	Terminal	B/O	B - 24V
RED	Terminal	B/O	W - 24V
CPI	CPI to Ethernet	BLUE	CPI
Heater Control		Black	
BLACK	6-pin	BLACK	G
WHITE	6-pin	WHITE	C8
BROWN	6-pin	GREEN	C7
Thermocouple		Brown	
RED	Type E	RED	SE4 - L
PURPLE	Type E	PURPLE	SE3 - H
Reference Thermistor			SE1 - Red
			Shield - Ground
			VX1 - Black
Shed			
Rain Gauge		SHIELD	
BLACK		BLACK	P1
WHITE		WHITE	Ground
GPS			
SHIELD			
BLACK			
RED			
BLUE			
YELLOW			
WHITE			
GRAY			