MAE 579 Wind Energy - Fall 2022

Homework 3 Report

Dr. Ronald Calhoun

By: Shubhvarata Dutta, Chethan Umashanker Reddy, Ethan Wyngard

Abstract

This report aims to analyze the processing methods of Velocity Azimuth Display (VAD) to approximate both virtual and real world wind velocities. By artificially generating wind data and modifying it to replicate physical data, it is possible to gain a deeper understanding of the inner mechanisms of how virtual lidars operate the vector mathematics associated with it. Furthermore the results of these tests have been synthesized to better interpret real implications when dealing with implementation of wind energy projects, including the presence of nocturnal jets and other meteorological and boundary layer phenomenon.

Analysis

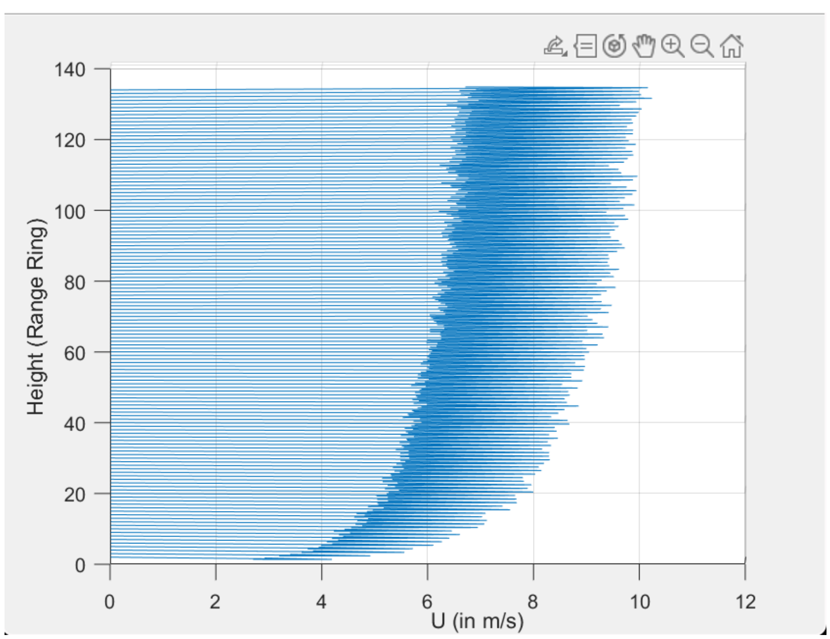
The first step in this process was to initially generate artificial wind vectors to pass through the VAD code. This was done so by creating u and v component vectors and applying a log law and spiral to them in such a way that it mimics real data. Additionally a ‘random kick’ was implemented in order to simulate the invariability in real wind profiles. Doppler lidar works by producing conical scans at varying heights in an effort to view the same wind vectors from all angles to reconstruct them in three-dimensional space. This is done by first finding the unit vector in the ‘look’ direction along the path of the scan and taking the dot product with the horizontal velocity vectors. This yields a set of radial velocity vectors, which will be the input to the calculation. Data from one full range ring is passed through the VAD code to produce a single velocity vector, which is then iterated to obtain all the u and v components of the radial velocity. This is calculated by means of the least squares method to solve the system of linear equations. In this case, it can be simplified to the form $Ax = b$, where b contains the radial velocity vectors, and x is the unknown u and v components, and A is a matrix of calculated values given by the following, where the last component is approximated to be 0:

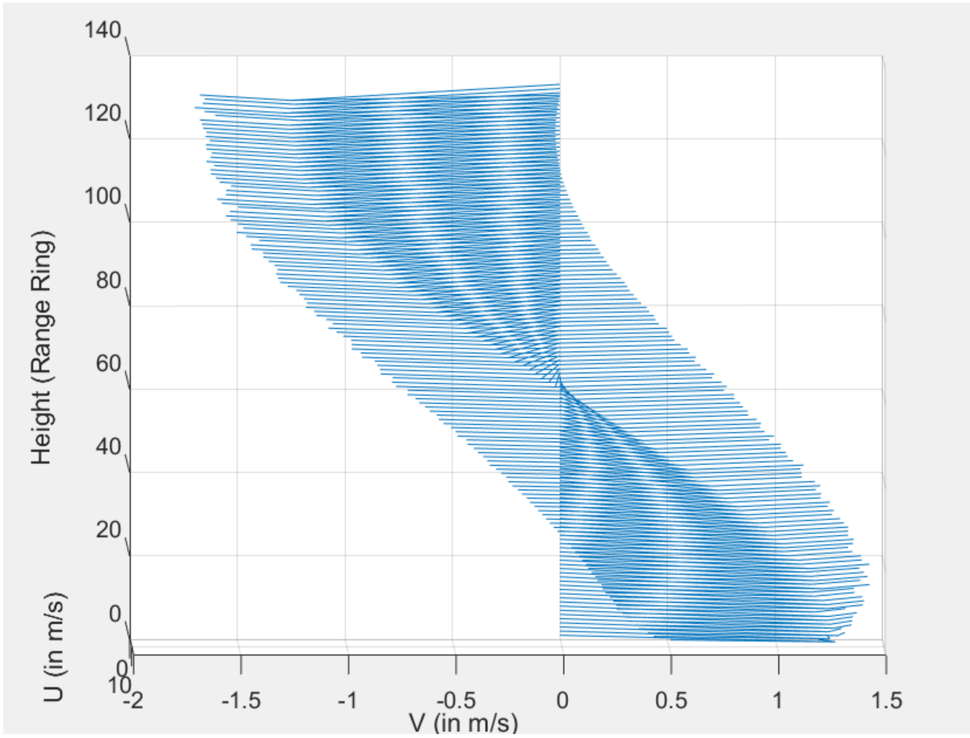
$$\mathbf{A} = \begin{pmatrix} \sin(\alpha_1)\sin(\phi) & \cos(\alpha_1)\sin(\phi) & \cos(\phi) \\ \sin(\alpha_2)\sin(\phi) & \cos(\alpha_2)\sin(\phi) & \cos(\phi) \\ \sin(\alpha_3)\sin(\phi) & \cos(\alpha_3)\sin(\phi) & \cos(\phi) \\ \ldots & \ldots & \ldots \\ \sin(\alpha_n)\sin(\phi) & \cos(\alpha_n)\sin(\phi) & \cos(\phi) \end{pmatrix}.$$

Figure 1: Paschke

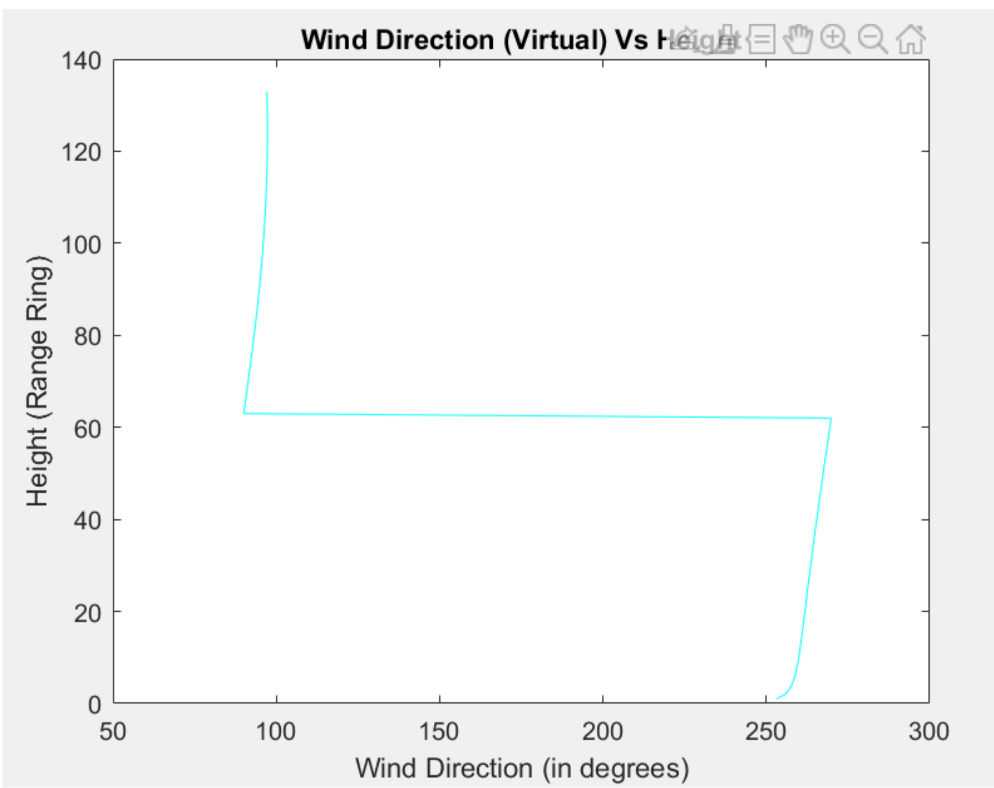
Once these vectors are obtained, it is possible to generate multiple plots including horizontal velocity vs height, wind direction vs height, and mean profiles of both. These plots can be compared between the artificial data and real data in order to determine nuances and characteristics of the two. Furthermore, with this data, spanning multiple instances of scanning, one can create an animation that displays how wind speed and time vary throughout the day.

Results

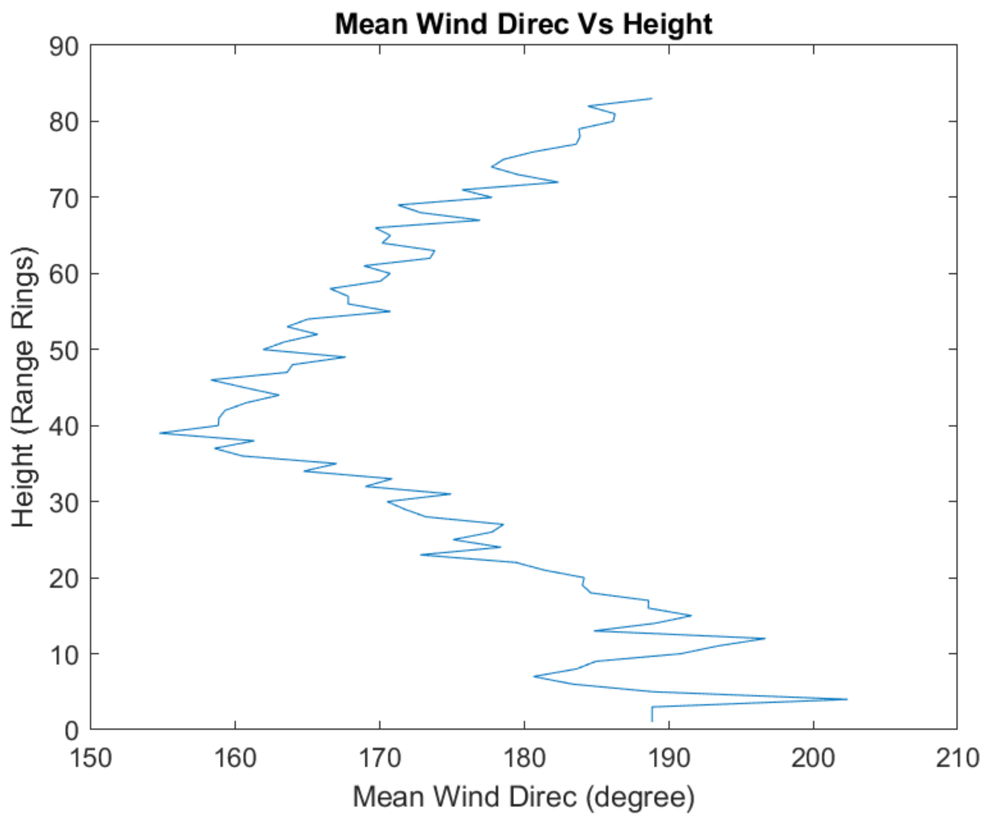


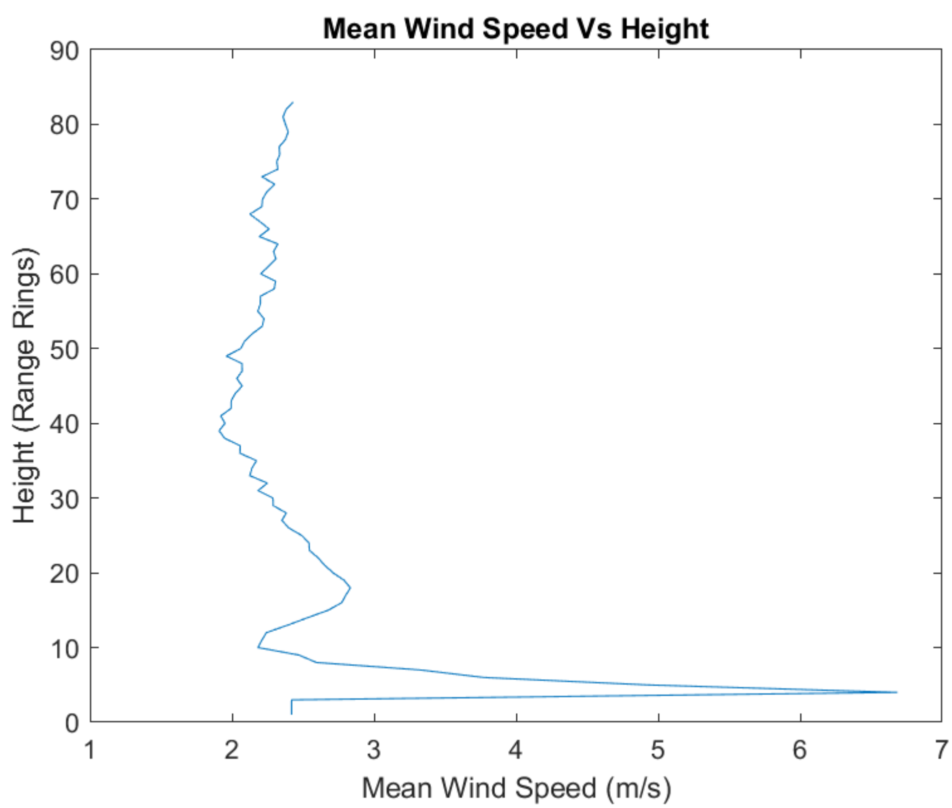


140
120
100
80
60
40
20
0
Height (Range Ring)
U (in m/s)
0
10
-2
-1.5
-1
-0.5
0
0.5
1
1.5
V (in m/s)



Wind Direction (Virtual) Vs Height
140
120
100
80
60
40
20
0
Height (Range Ring)
50
100
150
200
250
300
Wind Direction (in degrees)


Mean Wind Direc Vs Height
Height (Range Rings)
Mean Wind Direc (degree)
0
10
20
30
40
50
60
70
80
90
150
160
170
180
190
200
210


Mean Wind Speed Vs Height
Height (Range Rings)
Mean Wind Speed (m/s)
0
10
20
30
40
50
60
70
80
90
1
2
3
4
5
6
7

Synthesis

There are many different types of jet streams, all varying in wind speed, height, and time of occurrence. Nocturnal jets are classified as Boundary Layer Jets which generally propagate in clear night skies and are perpetuated through the next day. These can be found a couple hundred meters above the ground and often contain strong wind shears and diurnal variation (Davies, 2000). They are quite prevalent, which is why they are heavily considered when dealing with aviation endeavors, pollutant transportation, and storm propagation. They should especially be given thought when working with wind power generation, as they can occur in the same atmospheric level as the turbines. This is clear when viewing the wind speed vs height graph because there is a dramatic difference in the speed closer to the ground compared to higher in the troposphere. Additionally, they are known to develop over both land and sea, so they are relevant to offshore wind projects as well. Given this, the vertical wind shear is crucial when regarding the potential rotor swept area, because it will dictate the height at which it would be advantageous to implement the turbine. From viewing the animation of the wind speed versus height, it is also evident that the profiles change as the day passes, not unlike the Stull boundary layer daily evolution image. Furthermore, the wind direction varies greatly over the course of the day. This is indicative that the data was collected over a relatively flat location where the wind can blow freely. This is in contrast, for example, to the Turkana location which had very specific earth formations that directed the wind one way throughout the day. The direction of the wind can also change with the time of day.

References:

Davies, P. A. (2000). *Development and Mechanisms of the Nocturnal Jet*. Development and mechanisms of the nocturnal jet. Retrieved October 18, 2022, from https://rmets.onlinelibrary.wiley.com/doi/pdf/10.1017/S1350482700001535

Ishwardat, N. K. S. (1970, January 1). *Radar based horizontal wind profile retrieval techniques: DFT applied to scanning Doppler radar measurements*. TU Delft Repositories. Retrieved October 19, 2022, from https://repository.tudelft.nl/islandora/object/uuid%3Aa659654b-e76a-4513-a656-ecad761bdbc8

Paschke, E., Leinweber, R., & Lehmann, V. (2015). *An assessment of the performance of a 1.5μm doppler lidar for ...* Retrieved October 20, 2022, from https://d-nb.info/1142269299/34

By- SHUBHVARATA DUTTA / Ethan Wyngard / Chethan Umashanker Reddy

Contents

MAE579 - Wind Energy HW-3

```
clear all;
close all;
clc;

% % Wind Energy HW 3

% % Import wind data

    load('Data_for_VAD.mat');
    range = Data.range;
    az = Data.az;
    el = Data.el;
    rv = Data.rv;

% % Generate random velocity vectors


    V = 20 + zeros(133,1);     % Assume wind vel in x-direction const. = 20m/s

    x = 1:133;
    y = 1:133;
    z = 1:133;

        z0 = .03 ;                 % Roughness length (Assumed: Flat Plains)
        ustar = (.075) .* V ;      % Friction velocity, generally ustar/u = .05 to .1
        k = .4 ;                   % Von Karman Constant
```

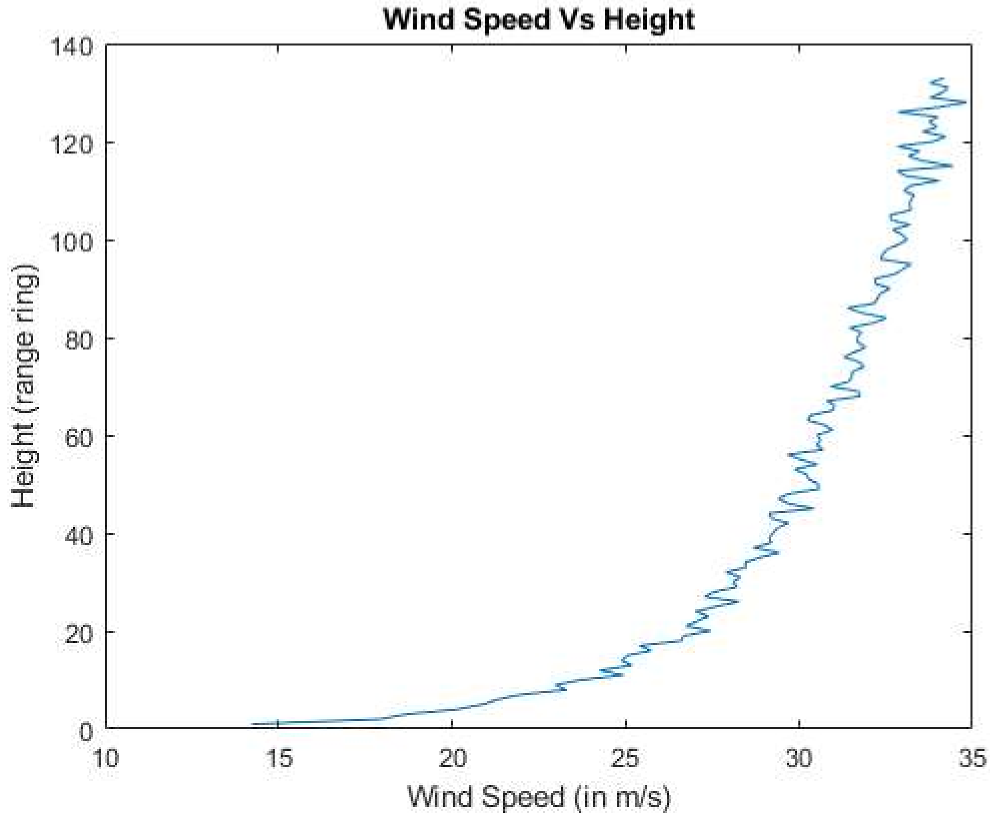
Applying log law

```
    u = (ustar./k).*log(z'./z0);    % u-component wind vel. with log-law


    v = zeros(133,1);

    for i=1:length(z)
        v(i,1) = 4*cos(0.025*i);
    end

    w = zeros(133,1);

    % Random kick

    rk = normrnd(.075,.015,133,1);
    for i = 1:length(rk)
        u(i) = u(i) + u(i)*rk(i);
    end

    for i = 1:length(rk)
```

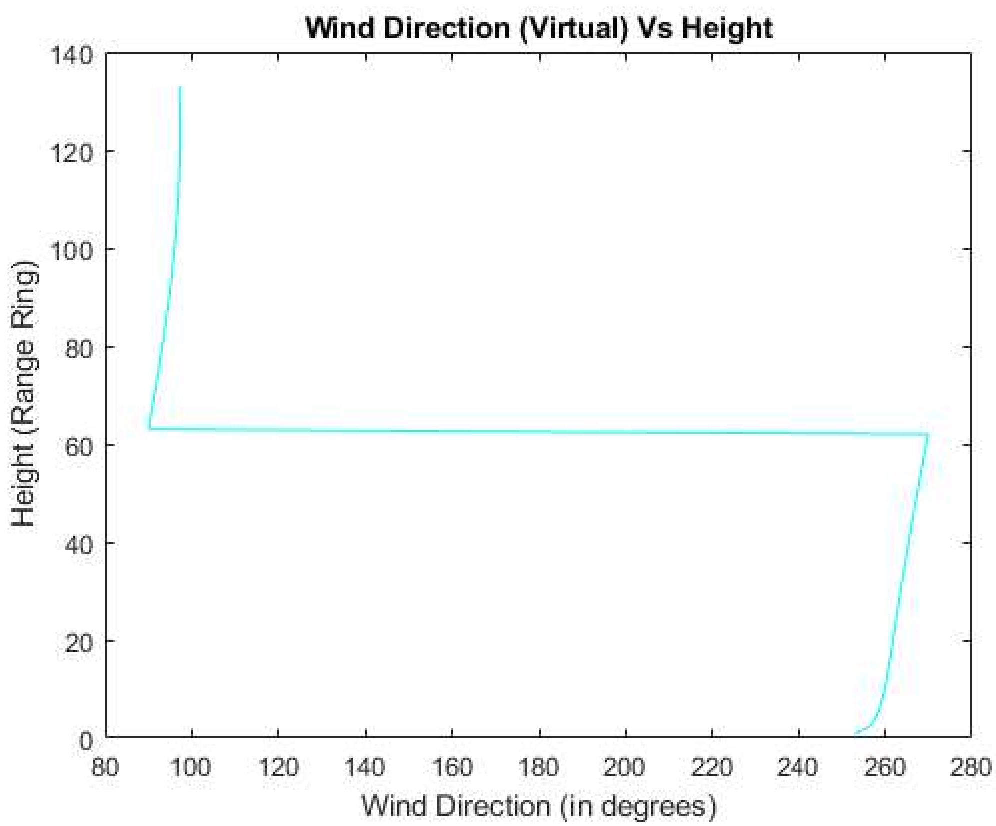
```
        v(i) = v(i) + v(i)*rk(i);
    end

    V_resltant = sqrt(u.^2 + v.^2);

    figure
    plot(V_resltant,z)
    xlabel('Wind Speed (in m/s)');
    ylabel('Height (range ring)');
    title('Wind Speed Vs Height');

    theta = zeros(133,1);

    for i = 1:133
        if u(i,1) > 0 && v(i,1) > 0
            theta(i,1) = 270 - atand(abs((v(i,1)/u(i,1)))) ;
                    else if u(i,1) > 0 && v(i,1) < 0
                        theta(i,1) = 90 + atand(abs((v(i,1)/u(i,1)))) ;
                        else if u(i,1) < 0 && v(i,1) < 0
                            theta(i,1) = 270 + atand(abs((v(i,1)/u(i,1)))) ;
                            else if u(i,1) < 0 && v(i,1) > 0
                                theta(i,1) = 90 - atand(abs((v(i,1)/u(i,1))));
                            end
                        end
                    end
        end
    end


    figure
    plot (theta', z, 'c');
    xlabel('Wind Direction (in degrees)');
    ylabel('Height (Range Ring)');
    title('Wind Direction (Virtual) Vs Height')

    x1 = zeros(size(z));
    y1 = zeros(size(z));

    figure
    quiver3(x1,y1,z,u',v',w')
    xlabel('U (in m/s)');
    ylabel('V (in m/s)');
    zlabel('Height (Range Ring)')

    whos
```

```
  Name                Size                Bytes  Class     Attributes

  Data                1x516           182483064  struct
  V                 133x1                  1064  double
  V_resltant        133x1                  1064  double
  az                 83x133               88312  double
  el                 83x133               88312  double
  i                   1x1                     8  double
  k                   1x1                     8  double
  range              83x133               88312  double
  rk                133x1                  1064  double
  rv                 83x133               88312  double
  theta             133x1                  1064  double
```

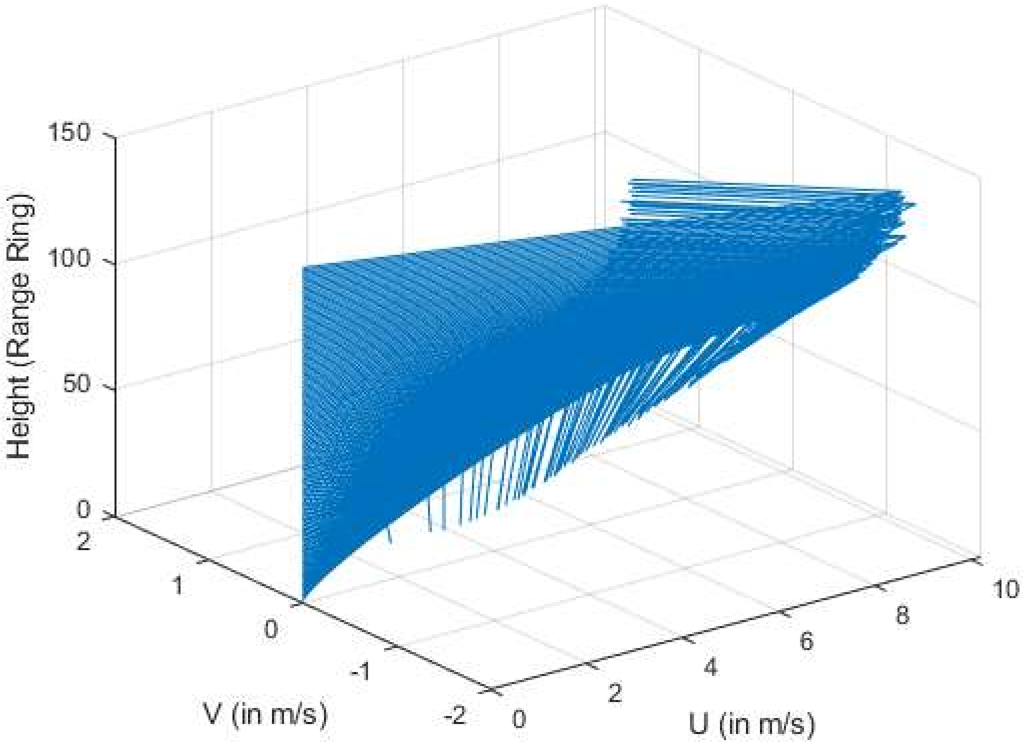
```
  u               133x1                   1064  double
  ustar           133x1                   1064  double
  v               133x1                   1064  double
  w               133x1                   1064  double
  x                 1x133                 1064  double
  x1                1x133                 1064  double
  y                 1x133                 1064  double
  y1                1x133                 1064  double
  z                 1x133                 1064  double
  z0                1x1                      8  double
```



Wind Speed Vs Height
Height (range ring)
Wind Speed (in m/s)
0
20
40
60
80
100
120
140
10
15
20
25
30
35



Wind Direction (Virtual) Vs Height
Height (Range Ring)
Wind Direction (in degrees)
0
20
40
60
80
100
120
140
80
100
120
140
160
180
200
220
240
260
280



Published with MATLAB® R2021b

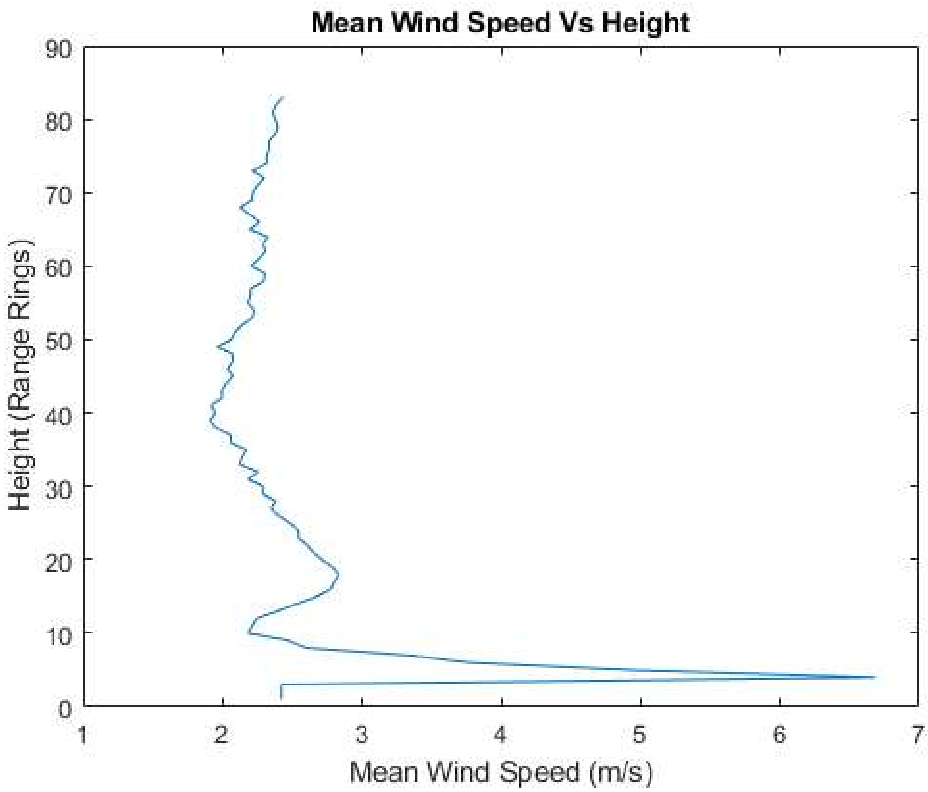
Contents

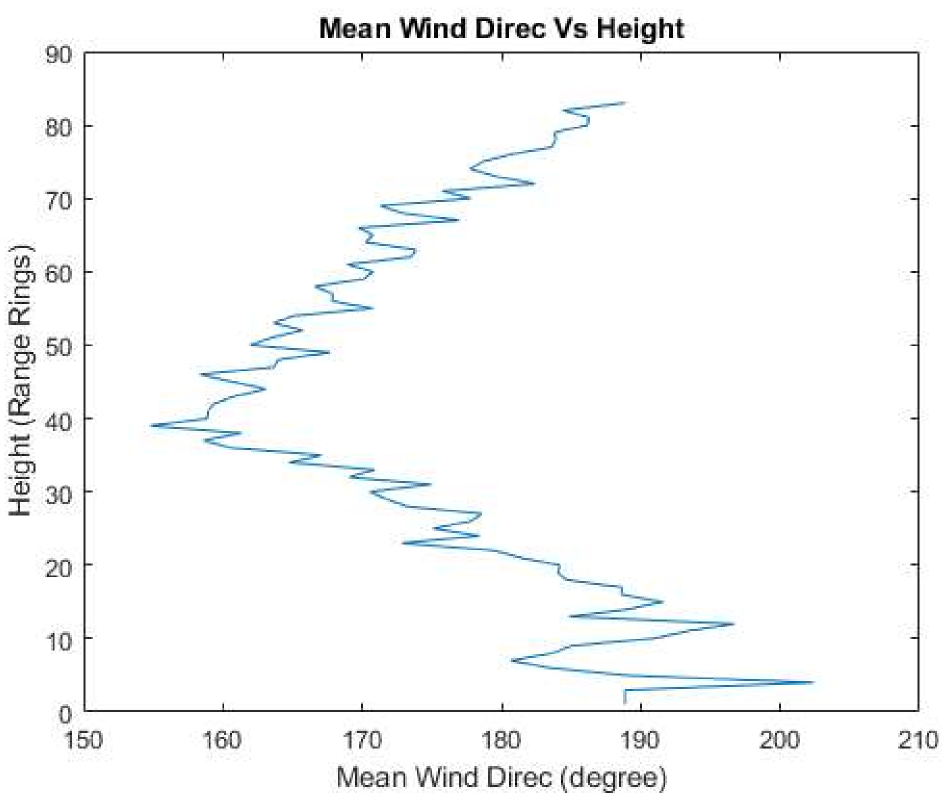

```
clear all;
close all;
clc;

% % Wind Energy HW 3 - VAD Algorithm On Given Wind Data

% % Import wind data

    load('Data_for_VAD.mat');

    %     Rv = U*sin(phi)*cos(theta) + V*cos(phi)*cos(theta);
```

VAD Calculation

```
    clc;

    A = zeros(133,2);
    B = zeros(133,1);
    T = length(Data);
    U = zeros(83,T);
    height = 1:83;
    theta = zeros(83,T);


    for t=1:T

        range = Data(t).range;
        az = Data(t).az;
        el = Data(t).el;
        rv = Data(t).rv;
        rvnan = ~isnan(Data(t).rv);

        for i = 1:size(Data(t).rv,1) % 1 to 83
        %     i=24;
            count = 0;
            for j = 1:size(Data(t).rv,2) % 1 to 133
                if rvnan(i,j) == 1
                    count = count+1;
                    A(count,1) = sind(az(i,j))*cosd(el(i,j));  %
                    A(count,2) = cosd(az(i,j))*cosd(el(i,j));
                    B(count,1) = rv(i,j);
                end
            end
            output = pinv(A)*B;
            vh = sqrt(output(1,1)^2 + output(2,1)^2);

            U(i,t) = vh;


            if output(1,1) > 0 && output(2,1) > 0
                theta(i,t) = 270 - atand(abs((output(2,1)/output(1,1)))) ;
                else if output(1,1) > 0 && output(2,1) < 0
                    theta(i,t) = 90 + atand(abs((output(2,1)/output(1,1)))) ;
                    else if output(1,1) < 0 && output(2,1) < 0
```

```
                    theta(i,t) = 270 + atand(abs((output(2,1)/output(1,1)))) ;
                    else if output(1,1) < 0 && output(2,1) > 0
                        theta(i,t) = 90 - atand(abs((output(2,1)/output(1,1))));
                    end
                end
            end
        end
    end
end



WindSpeed_Mean = mean(U,2);
WindDirec_Mean = mean(theta,2);

figure(1)
plot(WindSpeed_Mean,height)
xlabel("Mean Wind Speed (m/s)")
ylabel("Height (Range Rings)")
title('Mean Wind Speed Vs Height')

figure(2)
plot(WindDirec_Mean,height)
xlabel("Mean Wind Direc (degree)")
ylabel("Height (Range Rings)")
title('Mean Wind Direc Vs Height')
```

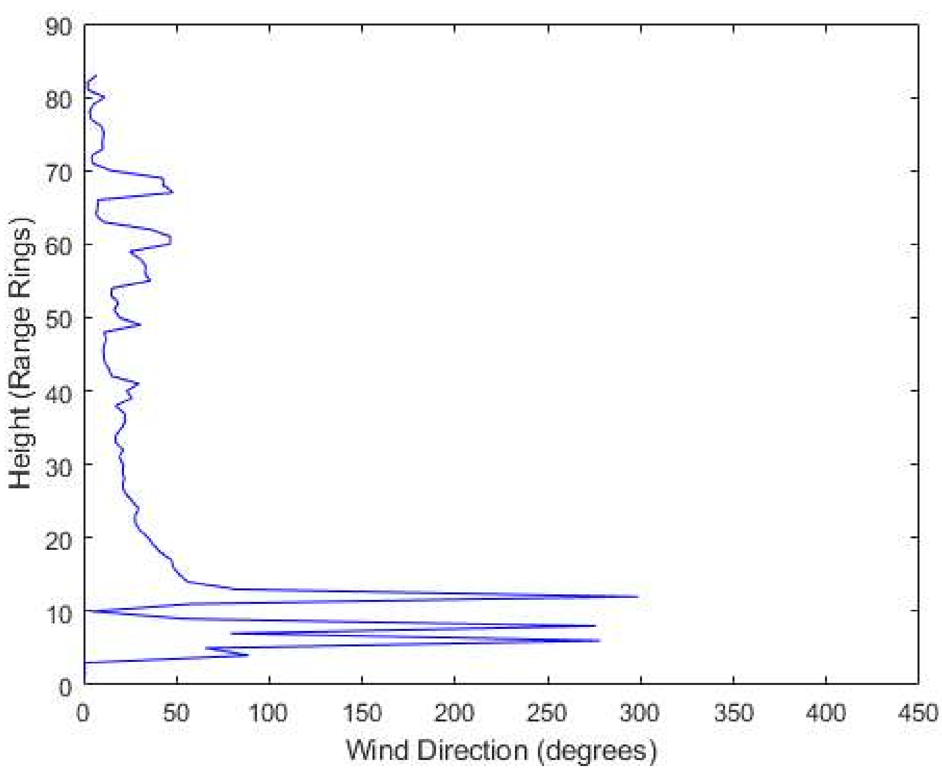




Animation

```
% Wind Speed
WindSpeed_vid = VideoWriter('E:\ASU Classes\MAE 597 Wind Energy\HW3\HW3Result\Wind_Speed_Profile.mp4','MPEG-4');
column=1;
while column<=516
    plot(U(:,column),height,'b');
```

```
    xlabel('Wind Speed (m/s)');
    ylabel('Height (Range Rings)');
    xlim([0 20]);
    pause (0.05);
    Vd1=getframe(gcf);
    open(WindSpeed_vid)
    writeVideo(WindSpeed_vid,Vd1)
column = column+1;
end
close(WindSpeed_vid)

%Wind Direction
WindDirec_vid = VideoWriter('E:\ASU Classes\MAE 597 Wind Energy\HW3\HW3Result\Wind_Direction.mp4','MPEG-4');
column=1;
while column<=516
    plot(theta(:,column),height,'b');
    xlabel('Wind Direction (degrees)');
    ylabel('Height (Range Rings)');
    xlim([0 450]);
    pause (0.05);
    Vd2=getframe(gcf);
    open(WindDirec_vid)
    writeVideo(WindDirec_vid,Vd2)
column = column+1;
end
close(WindDirec_vid)
```



Published with MATLAB® R2021b