Report on

“ThingSpeak Weather Station”

Submitted in partial fulfilment of the requirements for ***Sem IV***

IMAGE PROCESSING AND DATA VISUALIZATION USING MATLAB

Bachelor of Technology
in
Computer Science & Engineering

Submitted by:

Aishwarya Harinivas	**PES2UG19CS019**
R Sharmila	**PES2UG19CS309**

Under the guidance of

<Prof. Swati Pratap Jagdale>
<Assistant Professor>
PES University, Bengaluru

January – May 2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

TABLE OF CONTENTS

Abstract

ThingSpeak is an IoT analytics service that allows you to aggregate, visualize, and analyse live data streams in the cloud.

This project demonstrates how to build the weather station, configure the hardware, and interact with the weather data. We show how to retrieve the data from ThingSpeak, an IoT analytics platform. We then use histograms to visualize the data, interpolate missing data, fit trends to the data and compare current data to historical data.

Problem statement

Build your own portable weather station by analysing weather data in the cloud. Use ThingSpeak for retrieving data.

Module Description

- Retrieve data from ThingSpeak channel.
- Temperature, Humidity, Pressure, Rain, Windspeed, Wind Direction histogram
- Interpolation and contour for Temperature, Humidity, and pressure.
- Wind compass and Feather

High Level Design/Architecture

- **Retrieve data from ThingSpeak channel.**
  We read the weather data from ThingSpeak channel with ID 87179 and create variables to store the data read.
  The data read includes information about the temperature, humidity, pressure, rain, and wind.

- **Temperature, Humidity, Pressure, Rain, Windspeed, Wind Direction histogram.**
  We create 6 subplots in 2 by 3 plot. We create histograms to show the temperature, humidity, pressure, rainfall, wind speed and wind direction on different days.

- **Interpolation and contour for Temperature, Humidity, and Pressure.**
  We replace missing data by interpolation to keep the dimension of the array consistent. First, we create new x coordinate then we linearly interpolate as well as extrapolate the temperature, humidity, and pressure data.
  Then we find the maximum pressure and its index and create a surface fitting data. Then we plot the surface by making use of curve fitting toolbox. We then get the 2D view of this graph and mark location of maximum pressure.

- ## Wind compass and Feather

  We display the latest 10 wind directions using compass and feather plot. We start by converting direction data to radians. To create the feather plot we first remove missing data and any data with wind speed zero. Then we convert polar coordinates to cartesian and plot the feather.

  Next, we create the compass plot. First, we adjust the wind direction so that north is equal to zero degrees. Then we calculate the X and Y coordinates of wind speed on circular plot by calculating the cosine and sine components, respectively. We the plot the compass.

Implementation

```
%%% ThingSpeak Weather Station Data Analysis

%%Retrieve data from ThingSpeak channel
% Channel ID to read data from
readChannelID = 87179;
% Specify date range
dateRange = [datetime('March 7, 2016'),datetime('March 13, 2016')];
% Read data including the timestamp, and channel information.
[data,time,channelInfo] =
thingSpeakRead(readChannelID,'Fields',1:7,'DateRange',dateRange);
% Create variables to store different sorts of data
temperatureData = data(:,1);
humidityData = data(:,2);
pressureData = data(:,3);
rainData = data(:,4);
windSpeedData = data(:,5);
windGustData = data(:,6);
windDirectionData = data(:,7);


%%Temperature, Humidity, Pressure, Rain, WindSpeed, WindDirection
histogram
% Create subplots
figure % creates a figure window
% Temperature histogram
subplot(2,3,1) % Create 2-by-3 axis on the same figure, and work on the
first axis
histogram(temperatureData);
title(channelInfo.FieldDescriptions{1});
grid on

% Humidity histogram
subplot(2,3,2)
histogram(humidityData);
title(channelInfo.FieldDescriptions{2});
grid on

% Pressure histogram
subplot(2,3,3)
histogram(pressureData);
title(channelInfo.FieldDescriptions{3});
grid on

% Rain fall histogram
subplot(2,3,4)
histogram(rainData);
title(channelInfo.FieldDescriptions{4});
grid on
```

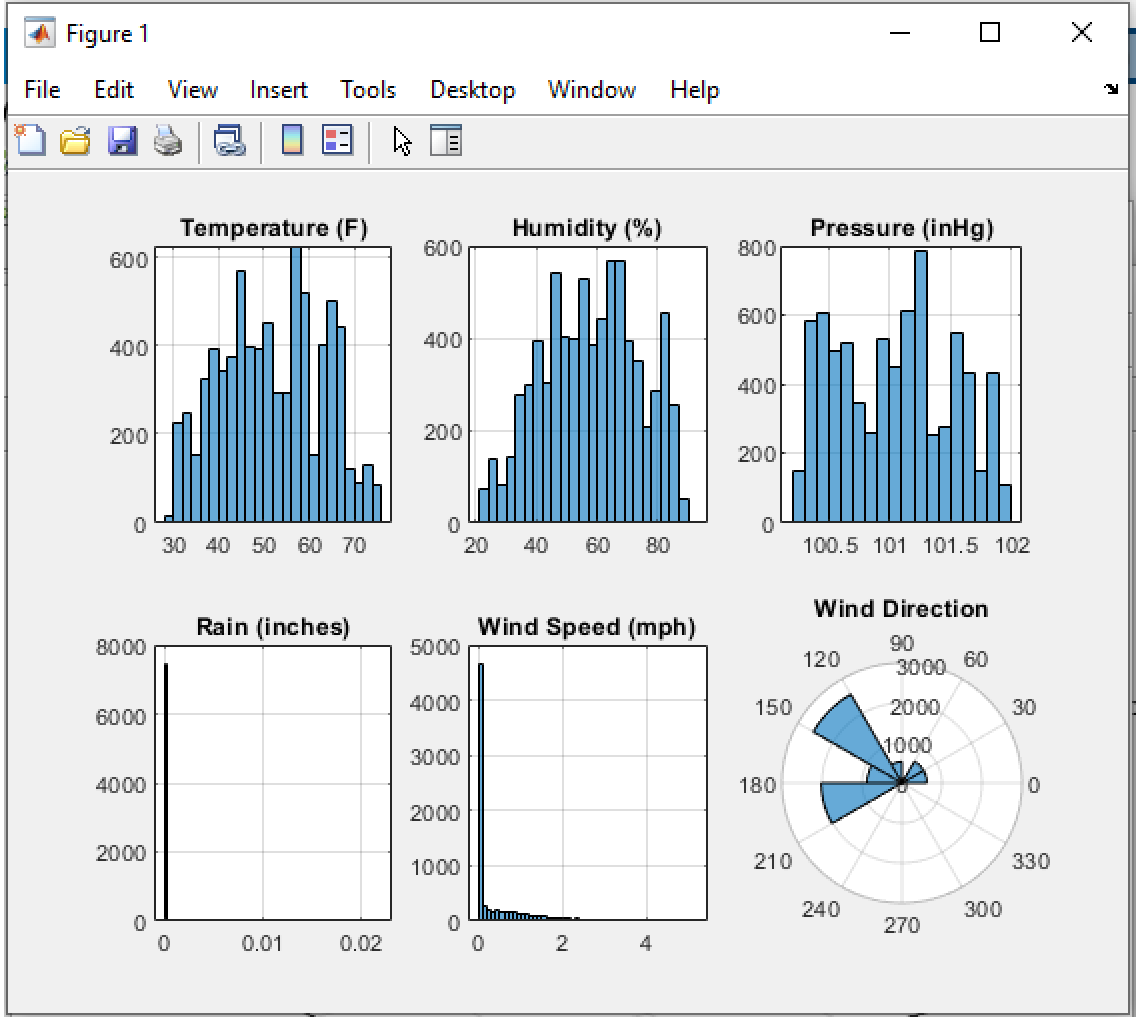
```
% WindSpeed histogram
subplot(2,3,5)
histogram(windSpeedData);
title(channelInfo.FieldDescriptions{5});
grid on

% Wind Direction histogram
rad = windDirectionData*pi/180; % Convert to radians
rad = -rad+pi/2; % Adjust the wind direction data to match map compass,
such that North is equal to 0 degree
subplot(2,3,6)
polarhistogram(rad,12) % Plot the wind direction histogram in a polar axis
title(channelInfo.FieldDescriptions{7})
ax = gca;
ax.View = [0 90]; % Rotate axis 90 degrees counterclock-wise such that
North is equal to 0 degree

%%Interpolation and contour for Temperature, Humidity and Pressure
% Replace missing data by interpolation, rather than removing the missing
data directly
% from the variable. This allows to keep the dimension of the array being
consistent
xNew = linspace(1,size(data,1),100)'; % Create new x coordinates
tNew =
interp1(temperatureData(~isnan(temperatureData)),xNew,'linear','extrap');
% Temperature interpolation. Extrapolation is applied here in case that
the last entry is NaN.
hNew = interp1(humidityData(~isnan(humidityData)),xNew,'linear','extrap');
% Humidity interpolation
pNew = interp1(pressureData(~isnan(pressureData)),xNew,'linear','extrap');
% Pressure interpolation

% Find the index of the max pressure
[pMax,idx] = max(pNew);

% Create surface fitting data
sf = fit([tNew,hNew],pNew,'linearinterp');

% Plot
figure
hsf = plot(sf,[tNew,hNew],pNew);
% Plot the surface with nodes. This plot function is provided in Curve
Fitting Toolbox.
%The output is an array of a surface object and a line object.
hsf(1).EdgeColor = 'interp'; % Change face edge color of the surface
hsf(1).FaceAlpha = 0.5; % Change the transparency of the surface
xlabel('Temperature')
ylabel('Humidity')
zlabel('Pressure')
title('Linear Interpolation Surface')

% 2D View with the location of max pressure
figure
hsf = plot(sf); % Plot the surface only
```

```
hsf.EdgeColor = 'interp'; % Change face edge color
hold on
plot3(tNew(idx),hNew(idx),pMax,'r.', 'MarkerSize',30)
% Plot the location of max pressure
text(tNew(idx)+2,hNew(idx)+2,pMax,['P= ',num2str(pMax),', T=',...
    num2str(tNew(idx)),', H=',num2str(hNew(idx))])
% Display the values at the location above
title('Contour of the Pressure')
xlabel('Temperature')
ylabel('Humidity')
grid off
view(2) % Set the view to 2D, i.e., observing the plot from top to bottom
along z-axis
hold off

%%Wind Compass and Feather
% Specify the latest n+1 wind directions to be displayed
n = 9;

% Convert to radians
rad = windDirectionData*pi/180;

% Create a feather plot
% Remove missing data and any wind speed with value 0
idx = (~isnan(rad)) & (~isnan(windSpeedData)) & (windSpeedData~=0);
% Convert polar coordinates to Cartesian. Note that dividing by the
maximum
% wind speed allows to scale the length of each arrow by its relative wind
% speed, rather than the wind direction.
[x,y] = pol2cart(rad(idx),windSpeedData(idx)/max(windSpeedData(idx)));
% Plot
figure
subplot(2,1,2)
feather(x((end-n):end),y((end-n):end)) % Plot the feather
xlim([0 n+2]) % Adjust the x-axis
ylim([-1 1]) % Adjust the y-axis
xlabel(['The last ',num2str(n+1),' wind direction']) % Add x lable
title('Wind Direction Changes')
grid on
ax = gca;
ax.YTickLabel = {}; % Hide the Y-Tick
ax.XTick = 1:(n+1); % Adjust the X-Tick for n+1 data

% Create a compass plot
% Adjust the wind direction to match map compass, such that North is equal
to 0 degree
rad = -rad+pi/2;
% Calculate the cosine component
u = cos(rad) .* windSpeedData; % x coordinate of wind speed on circular
plot
% Calculate the sine component
v = sin(rad) .* windSpeedData; % y coordinate of wind speed on circular
plot
% Plot
```

```
subplot(2,1,1)
compass(u((end-n):end),v((end-n):end)) % Plot compass
title('Wind Compass')
ax = gca;
ax.View = [-90 90]; % Rotate axis 90 degrees counterclock-wise such that
North is equal to 0 degree
```

Result snapshots

Temperature, Humidity, Pressure, Rain, Windspeed, Wind Direction histogram



Interpolation and contour for Temperature, Humidity, and pressure.

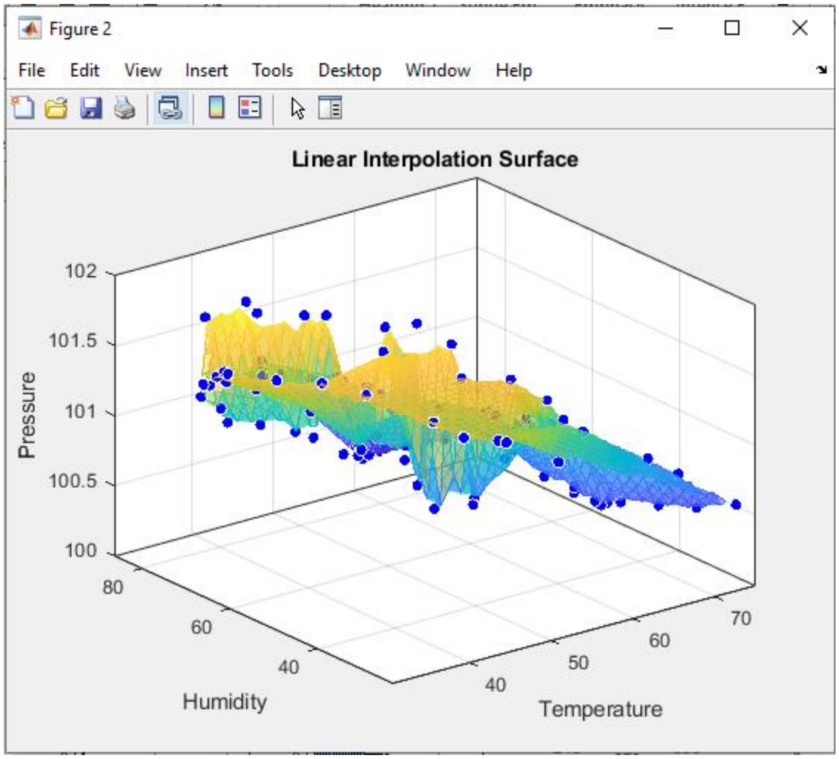


Figure 2
File Edit View Insert Tools Desktop Window Help
Linear Interpolation Surface
Pressure
102
101.5
101
100.5
100
80
60
40
Humidity
40
50
60
70
Temperature

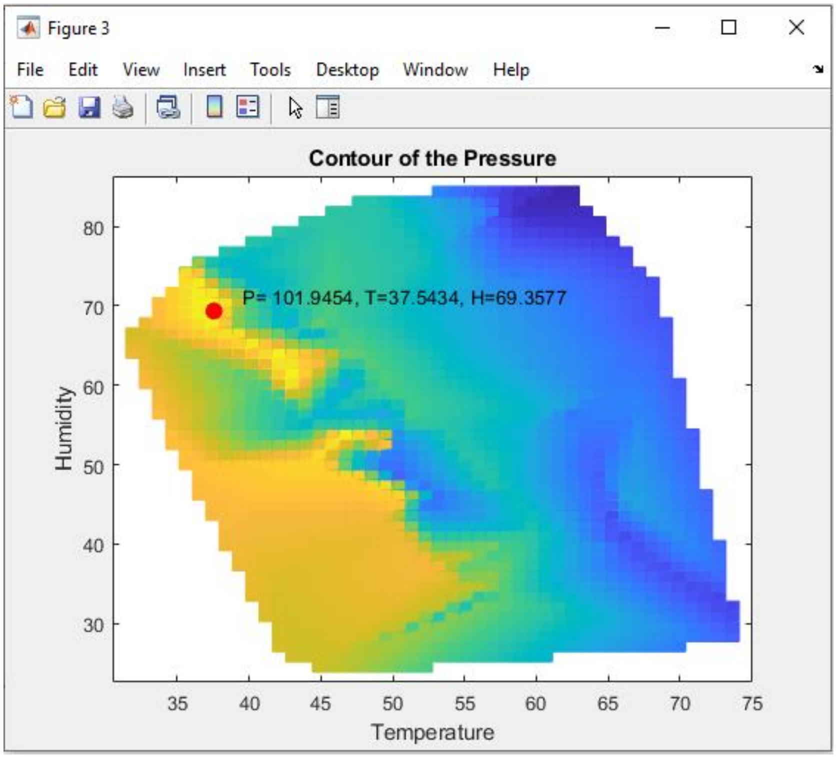


Figure 3
File Edit View Insert Tools Desktop Window Help
Contour of the Pressure
P= 101.9454, T=37.5434, H=69.3577
80
70
60
50
40
30
Humidity
35
40
45
50
55
60
65
70
75
Temperature

Wind compass and Feather

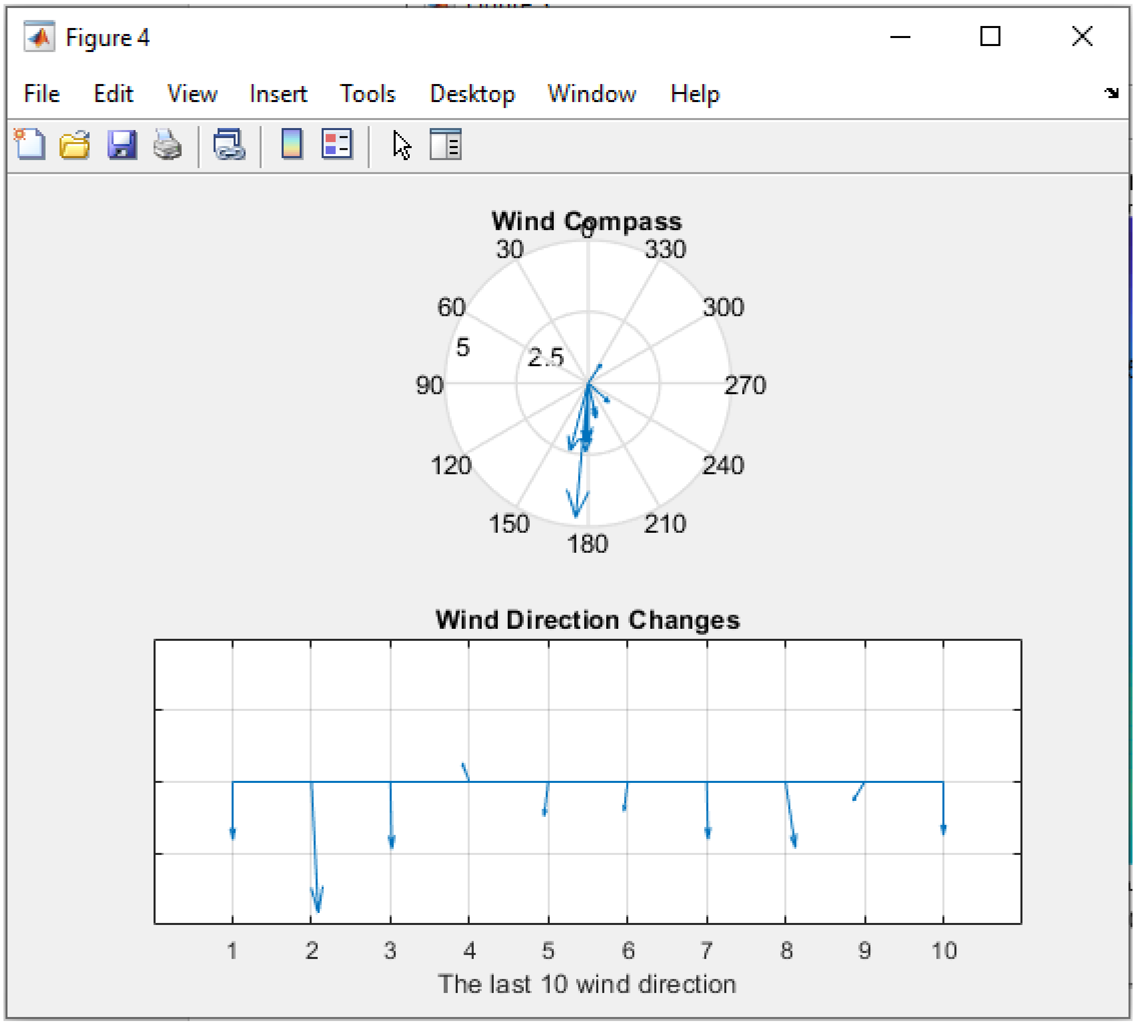


Figure 4
File
Edit
View
Insert
Tools
Desktop
Window
Help
Wind Compass
0
30
330
60
300
90
270
120
240
150
210
180
5
2.5
Wind Direction Changes
1
2
3
4
5
6
7
8
9
10
The last 10 wind direction