**ESTIMATING THE EFFICIENCY AND PERFORMANCE OF A SOLAR COLLECTOR**

**REPORT OF INDUSTRIAL TRAINING**

**At**

**IFB INDUSTRIES LIMITTED**

L-1, VERNA INDUSTRIAL ESTATE, VERNA

SALCETE, GOA 403722

Internship Duration: 13th May2019 to 2nd July 2019

A Report Submitted by

Yash Mahadik

3rd Yr. Undergraduate

Dept. of Ocean Engineering, IIT Madras

Chennai-36

under the guidance of

Mr. Rahul Patil

Deputy Manager R&D

IFB Industries Ltd.

Reviewed by:

Mr. Jayesh Wagh

Ms. Priyanka Sonawane

**TABLE OF CONTENTS**

**1. Objective**

**2. Project Description**

**3. Software Review**

**4. Results**

**5. Hardware Review**

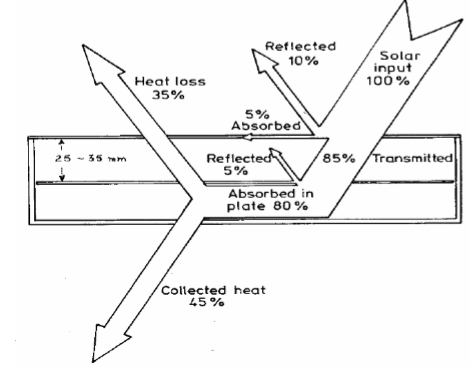
**6. References**

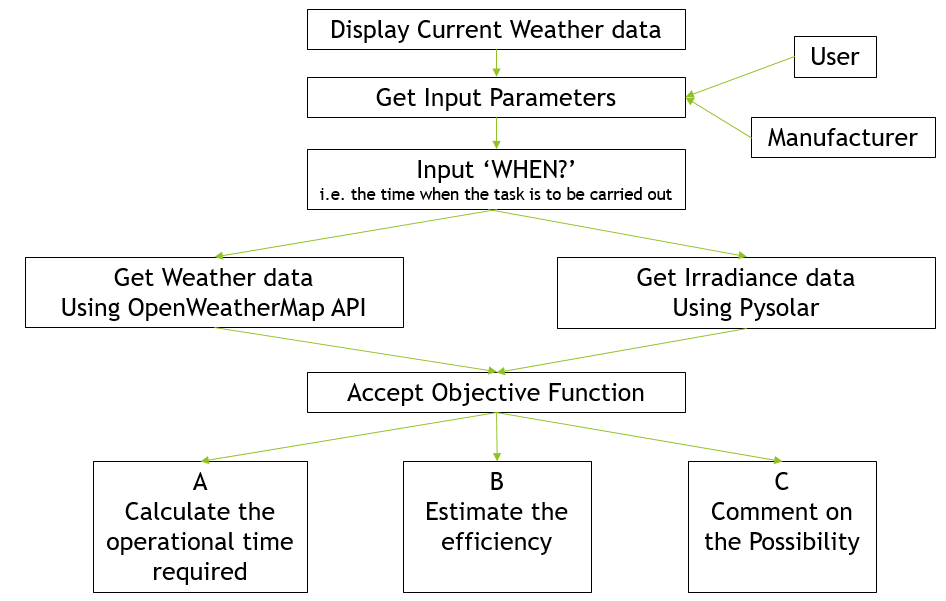
**1. Objective:**

Problem Statement:

* Required operational time
* Efficiency of the solar collector having known the ambient temperature
* Comment on the operational possibility considering other factors at the desired time

**2. Project Description**

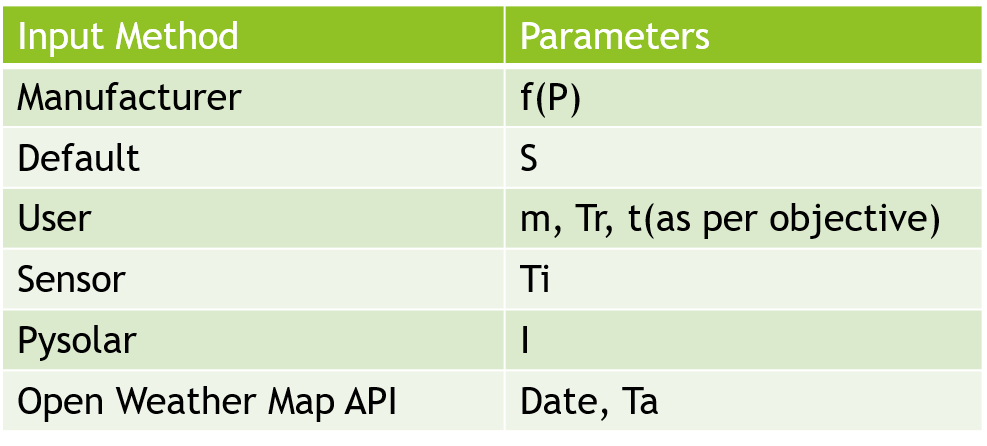
The aim was to achieve and eliminate the factor of dependency between environmental factors that affect the overall efficiency and the solar collector. We have solved the problem based on one assumption that the factor of the collector’s overall efficiency dependent on the ambient weather data is mutually exhaustive with the factor of the overall efficiency that accounts for all the other equipment (like pump, etc.) including the wear and tear.

We will understand further theory alongside the code given below

**# Install libraries**. (Here I have included all the libraries you might need. You can select only those that are necessary)

The basic framework is as follows:

a. We organize the data obtained using the Open Weather Map API by building the url and passing the content to a data organizer function.



Ta (): Ambient Temperature

Tr (): Required Temperature

Ti (): Inlet water Temperature

S (kJ/kgK): Specific heat capacity of water

I (Wm-2): Irradiance

A (m2): Area of the Collector plate

t (s): Time

By declaring the data variable global we can access it inside the main function later to get individual elements.

b. Theory with corresponding variables and objective functions.

= = f(P) = f

|  |  |
| --- | --- |
| Input Method | Parameters |
| Manufacturer | f(P) |
| Default | S |
| User | m, Tr, t(as per objective) |
| Sensor | Ti |
| Pysolar | I |
| Open Weather Map API | Date, Ta |

Ta (): Ambient Temperature

Tr (): Required Temperature

Ti (): Inlet water Temperature

S (kJ/kgK): Specific heat capacity of water

I (Wm-2): Irradiance

A (m2): Area of the Collector plate

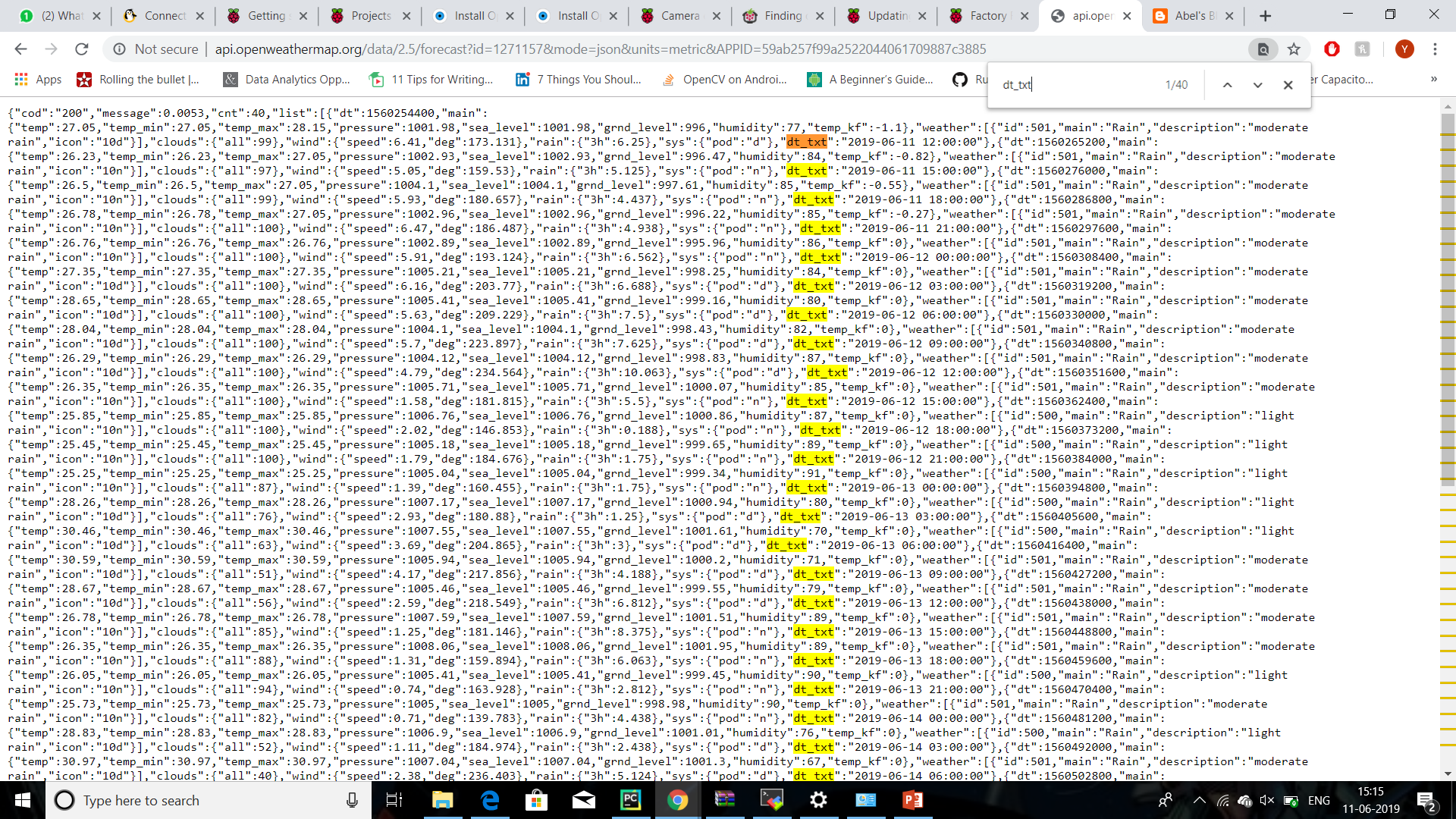
t (s): Time

m (kg): Mass of water



Based on the above equations and variables we can define our objective functions with necessary parameters.

c. Get Irradiance



We can see that the data provided from the Open Weather Map API is indexed with an interval of three hours between the time frames.  
  
  
For our purpose, we collect the data from the time frame which is closest to the time of operation.  
  
  
The date and time corresponding the above chosen index is used to obtain the irradiance of the sun at that point of time, which is then used to calculate the Available Energy for the Efficiency Calculation.

To get the solar intensity at a particular time of the day and considering our data constraints, we have included a simple function that takes the passes he nearest index value to the data organizer in the 1st step.



d. Main function



3. Results



4. Hardware Review

We have run the code discussed above on Python IDE platform JetBrains PyCharm Community Edition 2018.2.4.

For future practical usecases, we can get the inlet water temperature and mass of available water using sensors thus reducing user input.

5. References:

Supporting Theory

<http://www.lth.se/fileadmin/ht/Kurser/MVK160/Project_08/Fabio.pdf>

For obtaining API

<https://openweathermap.org/api>

For obtaining Irradiance

<https://pysolar.readthedocs.io/en/latest/>