Cloud Tracking for Short Term Irradiance Forecasting for PV Power Production

Introduction

Power production based on PV plants is a method that is currently increasing in popularity, but there are some issues that require some attention. The most important one for us, is the considerable decrements on the power production due to cloud events. These events cause high ramp rates on the power production curves which may be against the regulations at some countries. So what is needed, is a system that can predict the power production curve in a near future to allow the operators of the plant to take the necessary measures, such as starting another fossil-fuel based plant to ramp up, or disconnecting some cells to decrease the ramp rate early, to meet the regulation requirements. Our aim in this project is to produce the needed devices and computer software systems that would enable us to predict the power production curves in sub-hour ranges. For this purpose, the main components of the system are, the module that gives a baseline prediction on the clear-sky irradiance value, and a cloud tracking system that constitutes a camera system and the necessary tracking software to predict the drops on the production level.

For power production through PV panels, the only input comes from different components of the irradiance of the sun. At a point on the world, the measured total irradiance value at any time depends mainly on geographic position of the point, time of the year, weather conditions, and the amount of sun rays incident on the point. The listed first three factors are directly related to the extraterrestrial irradiation values measured outside the atmosphere, while the last two heavily depend on the atmospheric conditions. In fact, there are geometrical formulas that can accurately determine the effect of the first three, while for the last two we need a more complex formulation that depends on statistical methods. Even though, determining the effect of weather conditions are not easy, relying on the fact that the change in those are slow, there are some methods that predict the irradiance value using time series approaches. Nonetheless, these are ill-posed since the main factor that cause fluctuations on the irradiance value is cloud events that cannot be predicted by time series methods. It is because of this fact, that we need a more reactive sensor to observe the cloud events, and that is why we decided to use a camera.

In this report, the work throughout the last six months, along with brief information on the theoretical background behind, will be discussed. We are first going to explain the design of the system, and then explain all the subsystems that are used. We are also going to briefly explain the components that are prepared during the development process.

Theoretical Background

The power production of a PV panel is directly related to the incident irradiance on the field, and we want to accurately predict the irradiance value for a sub-hour period. There are several ways of doing what we want to do. In this part we are going to talk about baseline-drop methods, time-series based methods, and numerical methods.

Baseline-Drop Methods

The irradiance on a point is related to the extraterrestrial irradiation. The extraterrestrial irradiation is decreased during the propagation of the rays through the atmosphere. There are some models trained to predict that decrease during a clear-sky day. They take into consideration some averaged atmospheric factors and come up with some predictions on the level of irradiance. These models give us a baseline on the irradiance level for that time period, and the actual observed irradiance can only be lower than this baseline due to a drop caused by a cloud event. This group of methods predict try to predict the baseline accurately in several ways, and superpose the drop estimation that comes from the sky imaging module.

For the estimation of clear-sky irradiance levels, there are several approaches. Some try collect the clear-sky irradiance data of several locations and try to learn a model that is a function of geographical coordinates, while some others try to learn a model on the atmosphere characteristics, comparing the geometrical extraterrestrial irradiance with the measured ground irradiance, as a function of geographical conditions.

For the drop estimation, assistance of the camera system is needed. The camera software follows the clouds, and tries to predict the next state of them based on their previous states. As a result a 3D model of the field, sun, and the camera is built and the occlusion on the field can be predicted. The spatial and temporal resolution of the system depends on the way the clouds are modeled, the frequency with which the camera system updates with observations, and the tracking method that is used. For the tracking method, the alternatives are mainly correlation based methods on the whole image or on the individual areas, and model based tracking, where we build up individual cloud models and track each of them using a simple Kalman Filter. The quality of the drop estimation depends on how we characterize the cloud in terms of its type and the occlusion features of it. Modeling the cloud accurately from these perspectives can improve the prediction results.

This is the approach that we followed during the project development, more details on this will be provided throughout the report.

Time-Series Methods

As a scalar number, value of the incident irradiance on a point has a statistical trend, that can be tracked using a time-series based approach. The idea behind is simple, and it is that, the irradiance values follow a trend that does not change fast. So if we observe a drop on the irradiance measurements we predict that the drop is going to persist. The problem with this approach is that it is wrong. If there is a drop caused by a cloud event, it is probably a temporary drop and without a visual assistance in following the cloud event, there is no way of predicting the “return to normal” in the irradiance values.

There can also be some versions, that applies the time-series approach on a daily basis. These methods, tend to train a model that takes as input, some parameters related to the previous day, and gives an estimation of the hourly irradiance values for the next day. These methods may catch the big cloud events but still require some improvement provided by a camera system.

An idea that comes to our minds is that, the use of these methods for baseline irradiance level estimation can be tried as part of the project.

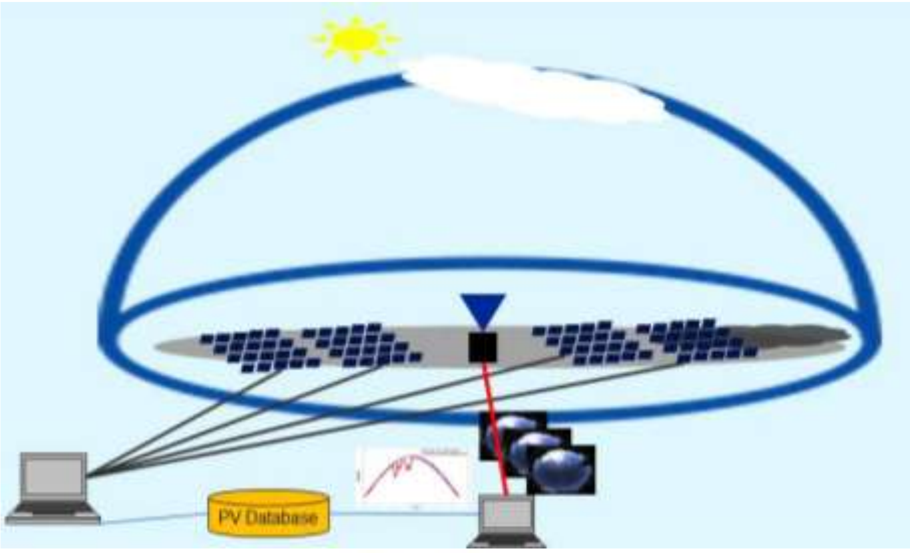
Numerical Prediction Techniques

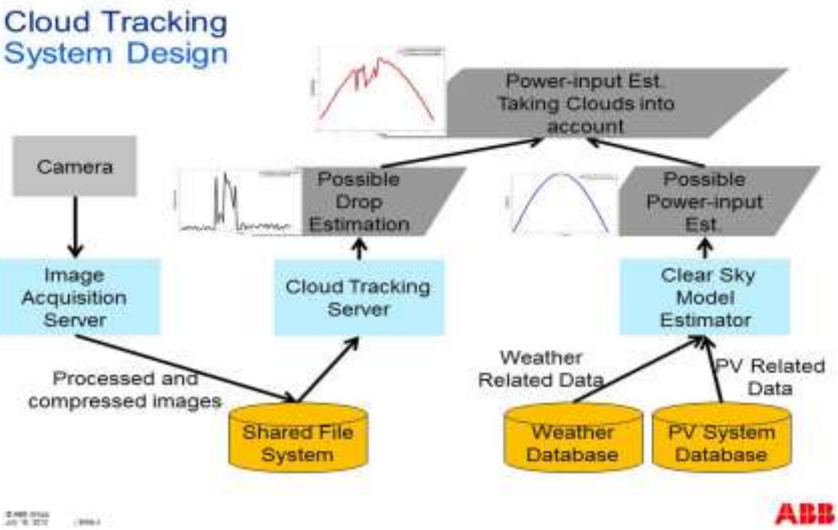
These are techniques that generate models that takes a large number of variables related to weather, location, time, and the geographical location of the plant and provide an estimation on the irradiance level.

In fact model generation is a very complex procedure, requiring very powerful machines, and a large quantity of data. Because of this reason, we do not even consider these methods for use.

System Overview

As we explained earlier, our approach is first predicting the baseline on the irradiance value using a model, and then predicting the possible drop that is going to happen in a near future using a camera system. So in effect our system contains two main parts, a camera system for cloud tracking and a clear sky model. In the below figures the a rough overview of the designed system along with the interactions between individual blocks can be seen. The camera is somewhere near the field directed upwards and is connected to the server that runs image acquisition and the tracking algorithm. This part of the system generates the drop estimations on the irradiance baseline using the cloud model it generates to get the placement of the shadow of those clouds with respect to the field. We also have another system that, using the stored data on the PV panel operation and weather conditions, estimates the baseline. Afterwards, the baseline and the drop estimation are superposed to get the irradiance estimation.





Sky Camera

We build the sky camera from the scratch. The requirements we ad were:

- The camera should be color camera, since according to many of the related work, the cloud identification relies on the histogram based features of the cloud and the sky.

- The camera should not introduce noise to the system. As for the tracking, the most important part is the identification of the clouds correctly, a low noise camera is a good idea.
- The camera selected should not be expensive. Preferably a webcam would be good.
- The camera should have the widest possible field of view. To be able to observe and sample a cloud accurately, and hence obtain a reliable prediction from the model.

For these requirements, we evaluated a wide catalogue of cameras, try to assess those according to their:

- Sensor type (CMOS,CCD)
- Sensor sizes
- Resolution
- Pixel sizes
- RAW to jpeg conversion technique used
- Bit-depth representation.

Afterwards, we contacted several suppliers for a possible quote for trial, and managed to get three camera systems from Fujinon-Basler Switzerland.
Our aim was to compare a high resolution camera that has a small pixel size, with the one that is low resolution and has bigger pixels. But then we could not have the time to try, and we used the camera with the highest resolution. The camera we used has a 1'' full-frame sensor with 4Mp resolution. The lens we use has a 185 degrees of field of view.

We also bought the housing of the camera from the supplier. The housing is a big one with dome opening for the camera to maximize the field of view. The problem we encountered with the housing system is that, because it is all sealed, it gets hot inside, and to cool it down, we needed to design a cooling system as well. The cooling system is a regular liquid based computer cooling system. We decided to go with that, we thought if we used another system because if we used we were afraid that there would be some condensation in the dome.

Irradiance Estimator

For the irradiance estimation, we implement several simple methods. The most important one is a daily time-series based method that uses a neural network for the estimation. It estimates the hourly irradiance values for the next day, taking:

- The trend and shape difference between the calculated extraterrestrial irradiation and the measured irradiance,
- The average temperature of the previous day,
- The peak temperature of the previous day,
- The day order number of the previous day.

We prepare the required data to input to the system, set up the neural network on MATLAB and do several trials on the training and testing, but we did not confirm the validity of the method.

Also we implemented six different extraterrestrial irradiation models to be used for trend estimation and neural network.

Cloud Tracking Algorithm

This is the algorithm that we use to track clouds. Flow of the algorithm is simple and for the n'th frame it is as follows:

- Get the n'th image from the camera
- Decide which pixels on the image correspond to clouds and which ones to sky,
- Get a binary image of clouds,

- Using a clustering algorithm, determine the contours of clouds in the binary cloud image
- Assuming that each contour defines one cloud, determine a representative point, called the centroid, for the cloud object
- Determine the bounding box of each cloud
- From the previous n-1 images, we should have some tracks some of which belongs to clouds that we see in the current image. Associate these tracks with the clouds that are detected in the current image. Before the association, since now we are on a new frame, and the time has passed, update the “current location” information of the tracks through prediction method of their Kalman filters. For the association use a Hungarian algorithm with a custom method for cost matrix implementation.(for the details of the cost matrix implementation, please refer to the code comments)
- Getting the associations,
  - Update the Kalman Filters of associated tracks,
  - Create new tracks for the observations that are not associated,
  - Update the statistics of the tracks that are not associated to any observation. Delete the old ones.

This is the rough explanation of the algorithm, for the details on how to achieve intermediary steps, please refer to the code with the comments.

Image Acquisition Tool

There are two versions of this tool. One version is configured for some duration and frequency of sampling, and started manually by the user. The other version is for long term acquisition missions. In that version, user configures the frequency of the capture, the start and end time for a day, along with some camera parameters, and start the code for once. Afterwards, code starts the capture at the defined start time with the given parameters, ends it at the specified end time, and repeats the same sequence every day.

This module is implemented using a multi-thread architecture. The main thread starts the program, initialize the folders for saving images captured, initializes the daily timer thread. The daily timer thread, works with a determined frequency and if the start time has passed, and the camera is not running, it initializes the camera and launches the acquisition thread. The acquisition thread takes pictures from the camera with the specified frequency, If the HDR is active, it merges the bunch of images that are taken, and saves them to a shared folder to be read by the image processing algorithm. Afterwards, when the daily thread realizes that the end time is passed, it terminates the acquisition thread and closes the camera. This cycle repeats every day.

Web-Server Reporter

We also wanted a handle for the system observers, so that they can see the state of the system. For this purpose, we provided a very simple web reporter. For this purpose we just export an image of the graphs that are shown by the cloud tracking algorithm, put those images on top of each other on a gif file, and then display that gif file in a HTML web page. For this we just extended the tracking algorithm with an extension to save gif images to a structured file system on a web server and then save an HTML file with the appropriate link to the public web folder. Please refer to the code files for the details.

Some Utility Tools

We also had to implement a lot of tools for the purpose of code reuse and efficient implementation of the system. Here I provide a list with some brief information on what they are.

- Kalman Filter : a generalized kalman filter implementation that provides several different kind of models for tracking.
- Video Player: to display the images for tracking in a video-like fashion.
- Model Player: to display the 3D model of the clouds, the plant, and the sun. Prepared for displaying the model continuously.
- Image Labeler: to be able to label the relevant part of information on images to enable us in pixel classification afterwards.

System Integration

As explained before, there are two main systems that works together. It works as follows:

- The image acquisition tool, captures images with the determined frequency, and saves them to a shared buffer folder.
- The tracking tool reads the image from the buffer folder and runs the algorithm that is explained under that subsection.

In case of tracking on a record of images:

- The tracking algorithm reads the folder that contains the images
- In order it processes images one by one, and mimics a similar behavior to that of the live version.

For the live version, the system becomes a big one that is mainly governed by two main threads, one running the tracking algorithm while the other running the image acquisition. Since they are sharing a buffer, the concurrency issues should be managed. These are all managed by the operating system level functions that are provided.

Conclusion

All in all, the components discussed are developed. Right now, our camera can automatically capture images with the frequency level that we request. We are currently in the period of generating the necessary database for the images, so that we can improve the performance of our algorithms later, using this bunch of data.