Software solution implemented on hardware system to manage and drive multiple bi-axial solar trackers by PC in photovoltaic solar plants

P. Visconti, P. Costantini, C. Orlando, A. Lay-Ekuakille, G. Cavalera

PII: S0263-2241(15)00437-6
DOI: http://dx.doi.org/10.1016/j.measurement.2015.08.024
Reference: MEASUR 3534

To appear in: *Measurement*

Received Date: 20 April 2015
Revised Date: 29 July 2015
Accepted Date: 18 August 2015

Software solution implemented on hardware system to manage and drive multiple bi-axial solar trackers by PC in photovoltaic solar plants

P. Visconti, P. Costantini,
C. Orlando, A. Lay-Ekuakille
Department of Innovation Engineering
University of Salento - Lecce, Italy
paolo.visconti@unisalento.it

G. Cavalera
Cavalera s.r.l.
Galatone, Lecce, Italy

***Abstract*—Solar energy is available almost everywhere but in some circumstances and locations it is necessary to optimize the dimensions of plants, avoiding large surface of PV modules installed on the ground, preferring modules located on solar trackers to increase the efficiency by at least 35%. However, even in this latter case, there are still margins for increasing the PV plant's efficiency. For this purpose, we have developed and tested an electronic system for controlling and driving bi-axial solar trackers of a PV plant, managed by a PC software application with a user-friendly graphical interface. The designed software is able to calculate the sun circadian orbit and consequently to move the solar panels in order to maintain the panel's surface always perpendicular to solar rays, improving the efficiency in energy production. In particular, the object of this work consists in optimizing an existing, designed by us, fully-hardware setup which didn't allow a simple and rapid plant management, completely replacing the Master electronic board with the designed software, so as to be able to communicate, by means of PC's RS232 serial port, with Slave electronic boards for tracker motors driving.**

Keywords- photovoltaic plant; solar tracker; efficiency; electronic equipment; renewable energy measurements.

I. Introduction

Project planning of a photovoltaic (PV) plant is preliminarily correlated to the choice between two different available types: fixed or motorized plant. In the fixed type, the incidence angle of the solar rays, on the panel surface, is conditioned by the panel inclination set during installation, the geographic latitude, the year period, the daytime and it cannot change. In the motorized type, exposure angle can change dynamically during day cycle with heliotropic movements that allow to keep the incident solar rays perpendicular to the photovoltaic panel surface, in order to maximize efficiency in electricity production. This system type is commonly said *solar tracker*; solar trackers can be classified according to freedom degrees, type of power supply for orientation mechanism and type of electronic control. Regarding to freedom degrees, solar trackers can provide a movement to the PV panels along a single or double axis and then they can be classified into single-axis trackers on tilt axis oron azimuth axis, roll trackers and bi-axial trackers. In this work we used a bi-axial solar tracker on tilt and azimuth axis (Fig.1), since, following the sun's circadian orbit, it exhibits major advantages in terms of efficiency gain regarding the energy production (35-40% more than a static system with respect to a 15-20% increase obtained with a single axis tracker) [1].

Figure 1. Example of solar panel with bi-axial (full tacking) solar tracker

Plots reported in Fig. 2 show a comparison between energy production in a fixed plant and in a PV plant with bi-axial solar trackers during daytime (Fig. 2a) and in different months of a year (Fig. 2b).

The greatest production benefits occur in the hours just after sunrise and before sunset. In fact, when solar radiation is more inclined on earth's surface, the solar tracker effect, that maintains the panel orthogonal to the solar radiation, gives the maximum improvement.

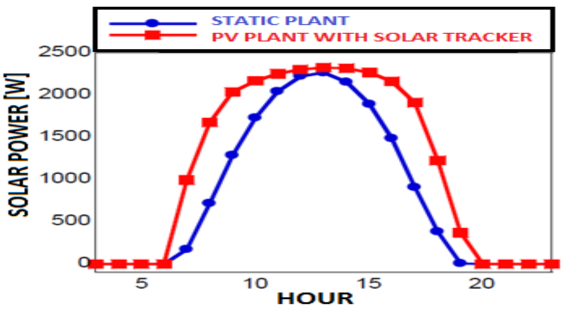


Figure 2a. Comparison between solar power production from a fixed photovoltaic plant (blue line) and from a dynamic photovoltaic plant with bi-axial solar tracker (red line) in different hours of a day

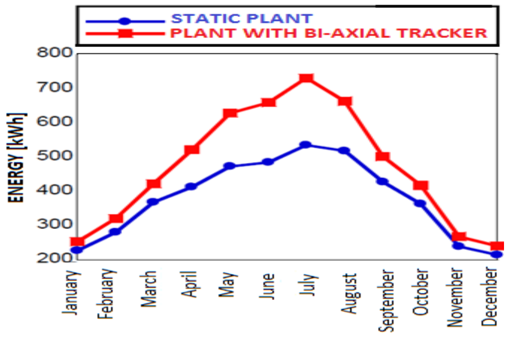


Figure 2b. Produced energy by a static photovoltaic solar plant (blue line) and by a dynamic plant with bi-axial solar tracker (red line) in different months of a year.

In this work a software, that runs on PC with master role, has been implemented in order to control slave electronic boards that drive motors for panels' biaxial movement. In order to obtain an algorithm on PC for driving, through the slave boards, the engines of solar trackers, mathematical formulas were used to calculate solar position during daylight hours in any day of the year, depending on the installation place of the photovoltaic plant, for maintaining panels always perpendicular to the sun. By a PC application with a simple interface, the common user can manage, control and reset the system parameters by PC software, without the technician intervention, however necessary in old non-informatized fully-hardware system. This user-friendly software interface for PV plant management represents the most innovative factor of the system presented in this work.

Gregor et al. [2] developed a similar system for biaxial solar tracking, based on Information and Communication Technologies (ICTs) and a GPS for the acquisition of sun's geographic coordinates in real time in order to improve the efficiency of the photovoltaic system. Bortolini et al. [3] implemented a system which follows Sun's apparent motion thanks to a proper combination of both a forward and a feedback control loop, using a completely automatic real-time monitoring platform based on LabView platform. Zubair et al. [4] designed a hardware system based on a microcontroller which automatically keeps the solar solar panels aligned with the sun in order to maximize efficiency. Khan et al. [5] developed another hardware microcontroller-based control system for solar tracking management based on light dependent resistors which are used as brightness sensors.

II. BLOCK DIAGRAM OF DESIGNED ELECTRONIC EQUIPMENT AND ITS ELECTRICAL BEHAVIOUR COMPARED TO PREVIOUS FULLY-HARDWARE SYSTEM

The goal of this work consists in optimizing an existing fully-hardware apparatus [6] for controlling and driving a solar tracker system, which didnot allow a simple and rapid plant management, completely replacing the Master electronic board with a software, running on PC with an appropriate application, able to communicate with Slave boards for tracker motors driving. Fig. 3 shows a block diagram of the previously designed system and fully functioning on several installed photovoltaic plants. This system consists of:

- C107 Master control board, manufactured by Cavalera srl, equipped with a Microchip microcontroller programmed with an algorithm able to calculate the sun position at the PV plant's latitude and longitude.
- C107 Slave actuation board, manufactured by Cavalera srl, that receives the inputs from master board on the solar position for driving correctly the panels' movement through the engines.
- GPS electronic system for signal acquisition from satellites of localization and time data.
- Anemometer for measuring the wind, used to prevent solar panels damage if they are subjected to excessive wind force, moving the panel surface horizontal and then parallel to the air flow, until the wind doesn't reduce its intensity.
- RS485 serial ports for Master-Slave communication.

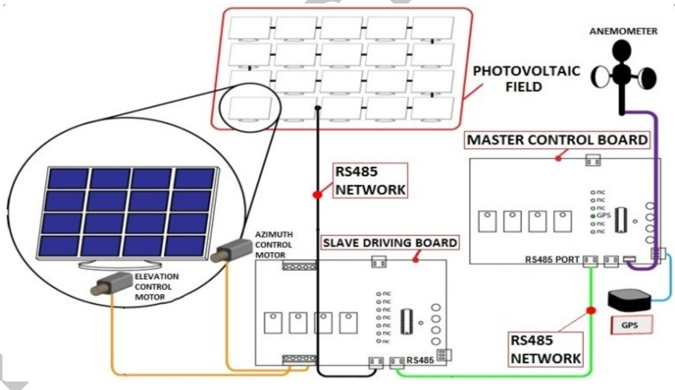


Figure 3. Block diagram of a photovoltaic plant with bi-axial solar tracker managed by old hardware solution with Master and Slave electronic boards

In the Fig. 4 are shown the Master (Fig. 4a) and Slave (Fig. 4b) electronic boards installed on previous hardware solution.

Figure 4a.C107 Master electronic board of PV plant with bi-axial solar tracker

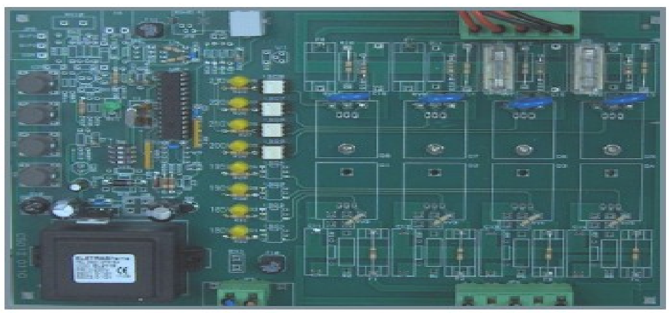


Figure 4b. C107 Slave electronic board of PV plant with bi-axial solar tracker

The designed new electronic equipment replaces Master board with a software running on PC, that controls, by RS485 port, all slave actuation boards (Fig. 5). The realized program is designed with a simple interface for both graphical and operational needs of a common user. This interface was developed through IDE Processing, a Java based environment GPL licensed. The features of the new improved system are:

- check on correctness of sent/received command strings and re-transmission in case of error up to 5 times.
- in previous solution it was not possible to set a starting time, for trackers' driving, different from sunrise or sunset; now, it is possible to anticipate or defer these default moments of system starting.
- all database's data on sun elevation and azimuth are always available on the application's research menu.
- if the PC is equipped with UPS (*Uninterruptible Power Supply*), the system maintains in memory the last position in case of power failure and it resumes its operation from the last saved coordinates, without resetting all trackers.

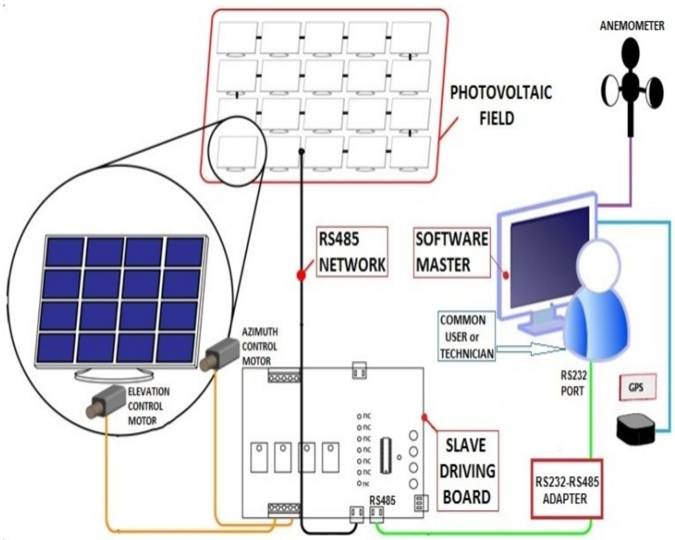


Figure 5. Block diagram of a photovoltaic plant with bi-axial solar tracker managed by new hardware and software solution with the implemented Master software on PC and Slave electronic board.

As inputs to slave boards, inductive sensors are connected in order to detect the run limit for azimuth and tilt (elevation) movement, useful for orienting correctly the PV string in sunrise state or in one of four cardinal positions in case of test command for equipment orientation. The run limit sensors are also useful for PV-structure repositioning in case of sudden absence of the main power after a black-out. In the event of undesired obstacle, the system can stop thanks to proximity sensors as depicted in Fig. 6.

The used inductive proximity's sensors are Wenglor IB040BM61VD devices powered with 24V DC voltage, IP67 degree of protection and switching distance of 4mm. These proximity's sensors allow, by means of appropriate cams located on the two axes of rotation, to establish the movement's limit of the engines in EAST, WEST, Horizontal and Vertical directions.

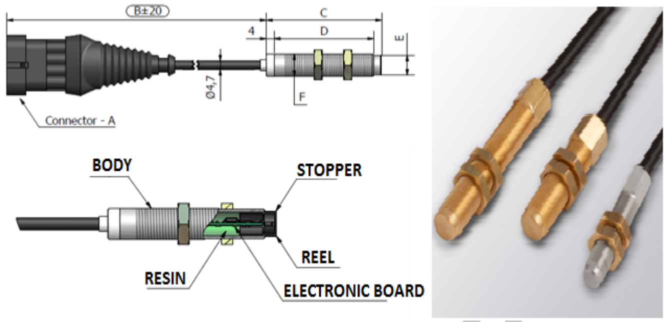


Figure 6. Proximity inductive sensors for stopping motor movement

III. Concepts of Spherical Astronomy and Mathematical Formulas for Solar Circadian Orbit

For locating a point on Earth, geographical coordinates of latitude and longitude are used as reference system. In earth coordinate system, the equatorial plane is chosen as a reference and the fundamental direction is the earth's rotation axis. A geographical meridian is defined such as a semicircle between the two poles and each meridian has its anti-meridian which completes the meridian circle in the opposite direction, so all meridians are equal. The parallels are circles formed by the intersection between any equator parallel plane and earth's surface, so they are smaller if their distance from equator grows. Parallels and meridians form a surface network, known as *geographical grid*, which allows us to identify the absolute position of a point. In order to find a precise parallel or meridian, geographical coordinates are defined. The geographical *longitude* Λ is the angular distance of a point from the prime meridian measured on the equator. It corresponds to the angle between the plane of the point's meridian and the plane of the prime meridian (Fig. 7). The geographic latitude φ is the angular distance between a point and the equator, measured along the meridian passing through the same point and corresponds to the angle between the vertical axis of the place and the equator plane. It can vary from +90° (*North Pole*) to -90° (*South Pole*) and the points along the equator are at a latitude 0° (Fig. 7).

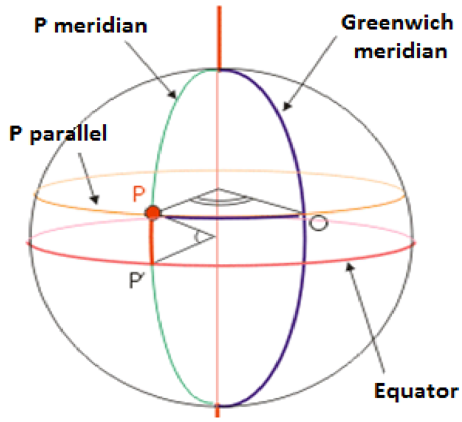


Figure 7. Geographic coordinate system with meridians and parallels

The designed algorithm for the solar tracker uses some mathematical formulas necessary for solar trajectory calculation. As known about the Earth's elliptical orbit around the Sun, the earth's distance, in meters, from the sun is given by:

$$d = 1.5\,10^{11}\left\{1 + 0.017\sin\left(\frac{360(n-93)}{365}\right)\right\} \quad (1)$$

where n represents the day calculated from 1th January. Earth also rotates around its polar axis employing 24 hours to make a complete rotation; due to the inclination of Earth polar axis with respect to the earth-sun orbital plane, equal to 23.45°, the sun is higher in the sky in summer than in winter, and daylight hours are longer in summer and shorter in winter. The sun's deflection angle from equator's straight line is called declination and is usually denoted by δ. For any given day n of the year, the declination can be found as:

$$\delta = 23.45°\sin\left[\frac{360(n+284)}{365}\right] \quad (2)$$

The declination can be related to zenith solar angle at noon, when the sun is at the highest point of its track in the sky (solar noon). The zenith complementary angle is called *elevation angle α*; it represents the angle between the horizon and the incident solar rays in a plane determined by zenith and the sun (figure 8).

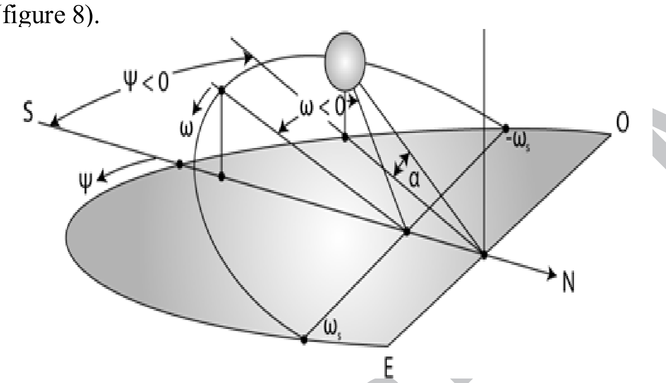


Figure 8. Graphic representation of elevation α, azimuth Ψ and time angle ω

The angular deviation of the sun from the south direction, can be described by the azimuth angle Ψ, which measures the angular position of sun in the East or West direction with respect to the south one; the azimuth angle is zero at solar noon and grows eastwards. The time angle ω is the difference between the noon and the desired daytime in terms of 360° rotation in 24 hours. In other words:

$$\omega = \frac{12-T}{24}360 = 15(12-T)^{\circ} \quad (3)$$

where T is the time of the day expressed respect to sun midnight. In solar time calculation, it is used the equation of time, which takes the following form:

$$EqOT = -10.1\left(\frac{360^{\circ}(2n+31)}{366}\right) - 6.9\sin\left(\frac{360^{\circ}}{366}n\right) \quad (4)$$

So, set the longitude Λ, the calculation of solar hour becomes:

(5)

Current_DayTime is the current time of day and depends on the time zone of plant's installation place; this information is available thanks to GPS connected to system's Master device (Master electronic board or PC software). As we can see from Eq. (5), two corrections were applied; the first one is due to Earth's speed in its revolution motion around the Sun, the second one is due to the difference in longitude with the central meridian of the time zone (4 minutes for degree of longitude), with positive sign in Western direction and minus sign in Eastern one. Specifically for the Italy southern area [7], *rif* parameter assumes the value of 15. In Fig. 9 is shown a representation of the elevation and azimuth in different months of the year at a latitude of 30° N.

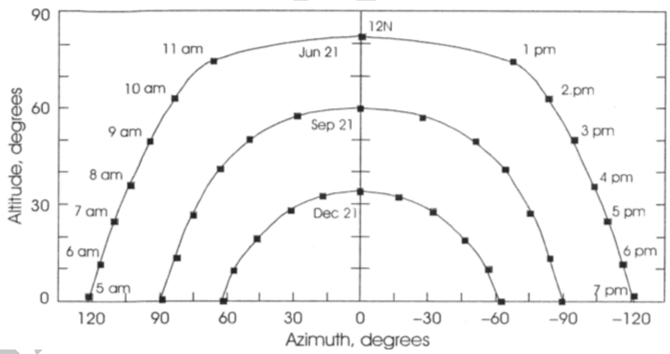


Figure 9. Graphic representation of elevation and azimuth in different months of the year

The curve shows the height of the Sun in the sky at different hours of day and for different months of the year. The table in Fig. 10 shows the required parameters used from realized software for the calculation of the solar position, in order to drive the slave boards for maintaining PV panels perpendicular to solar rays [3] [8].

SOLAR TRACKER PARAMETERS

Calculated day from January 1th 'n'	$m = (mounth + 9)\ \%\ 12$ $y = year - \frac{m}{10}$ $n = 365 \times y + \frac{y}{4} - \frac{y}{100} + \frac{y}{400} + \frac{m \times 306 + 5}{10} + (day - 1)$
Solar time 'T'	$T = \text{Current\_Daytime} + \frac{EqOT - 4 \times (rif - \Lambda)}{60}$
Time equation 'EqOT'	$EqOT = -10.1 \times \left(\frac{360° \times (2n+31)}{366}\right) - 6.9 \times sin\left(\frac{360°}{366} \times n\right)$
Time angle 'W'	$\omega = 15 \times (12 - T)°$
Declension "δ"	$\delta = 23.45° \times \sin\left[\frac{360 \times (n+284)}{365}\right]$
Elevation "α"	$\alpha = \arcsin(\sin(\delta)\sin(\varphi) + \cos(\delta)\cos(\varphi)\cos(\omega))$
Azimuth "ψ"	$\Psi = \arccos\left(\frac{sin(\alpha)\,sin(\varphi) - sin(\delta)}{cos(\alpha)\,cos(\varphi)}\right)$

Figure 10. Mathematical formulas used by software algorithm in order to calculate solar circadian orbit

IV. Flow chart of Firmware installed on master control board's PIC in previous hardware solution

The algorithm, implemented on PIC of master control board in the previous fully-hardware solar tracker system, could instantly calculate the sun position at the latitude and longitude of the installation site. Then the algorithm could drive up to two engines which are able to change the PV string's position, in order to increase its efficiency, for tracking the sun in its movement from east to west (azimuth motion) and in its elevation up to solar noon (tilt motion). In figure 11 is reported the flow chart relative to operating mode of master board's PIC which commands the slave boards so that the latter can properly drive engines of the solar trackers. After preliminary phase of GPS coordinates acquisition (latitude and longitude) and self-learning of the structure parameters, there's the acquisition of current time and check if it is greater or less than sunrise time. The process continues with calculation of daily parameters such as solar declination, equation of time, etc., necessary for the calculation of α and γ parameters relative to sun position. Then there is the execution phase of two engines and the check if sunset time is reached. If so, before returning to the acquisition time step, there is the east positioning of structure.

In the PC-based solar tracker system, the functionalities implemented by PIC on control board will be performed by the PC through an appropriate software and a dedicated application in order to be able to control by PC's RS-485 serial port all slave actuation boards.

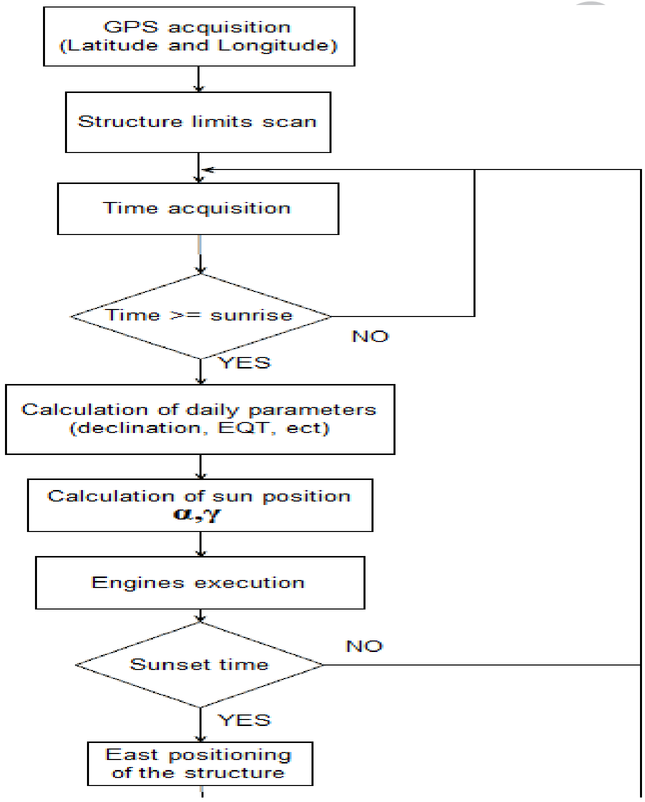


Figure 11. Solar tracking system's flow chart

V. Realized software on PC for solar position calculation and communication with slave boards

The software development environment used for calculation of solar position and subsequent creation of a related database, in order to establish a bidirectional communication with the slave boards through the PC, is IDE Processing. This software was chosen because it presents some important properties: the open character of compiler, a Java-based language and the potentiality to export it into different OS that support Java recompilation and finally the community support in relation with graphic designing of user interfaces. The most relevant characteristic of this software was a friendly interface, in order to allow all users (even the beginners) to understand the complexity of solar tracking without the know-how required, but with the only support of a graphical interface specifically designed to this purpose.

The first operation in order to design a graphical interface is the inclusion in the project of a library called *controlP5*, a processing GUI and controller library, which contains all base structures such as sliders, buttons, toggle and much more controllers usable in programming environment. Another relevant characteristic introduced with controlP5 was the separation of visual data in *Tabs*, used in a development environment, as Processing, with a single visual programming interface.

This property, in addition to determine a logic separation between inputs and outputs for any technical category, introduces a low time computation for graphic unit that processes all information that have to be shown each time. Lots of this information have been separated in different Tabs to order them, but even to reduce the CPU load, such as in tracking map shown in the Main tab (Fig. 12) and also to separate the different states of algorithm's information reporting or data evaluating (as in Slaves tab in Fig. 13). This approach has allowed to design a light interface, simple to use and with a graphic for the most part rendered by the CPU.

Two different software were designed for providing and then processing all necessary solar position data throughout the year independently from the GPS coordinates; the purpose was a self checked generation of data for the main software, this means even without GPS geographic coordinates. A first software has the only purpose of creating, in the first day of the year, a year-long database of values related to solar circadian orbit with time resolution of 10 minutes, during each day from sunrise to sunset. Then, these data are used as inputs for the second management software of the solar trackers.

The main software manages the communication with slave boards through RS-485 PC port, generating a pulse train whose duration establishes the correct movements of the solar tracker, corresponding to the database. In this way it is determined the time in which the tracker's engines have to remain on in order to allow the correct movement of PV trackers in the desired position. In addition to the duration of the pulse train, a bit of the command string determines if the tracker movement should be positive or negative (upward or downward).

In order to not do even minimal errors in solar tracker positioning, the system clock has to be synchronized; for this purpose a specific function of the software is able to extract the current time from the string information taken as input from GPS device. At the beginning, Processing was launched only as graphic environment, but after the community's effort it's evolved in lot of different solutions and drivers (even hardware), extending its range of application opportunities. It's the case of UART driver, present in class list with name *Serial*, which is a class for sending and receiving data using a standard serial communication protocol supported from different devices. This library has the flexibility to communicate with all MCU devices that have an UART port; the software writes and reads one byte at time that can be evaluated in two particular modes, sequential or for event.

The data sent and received from each solar tracker (Slave) are sequential with a fixed length, instead the data received from GPS (using NMEA communication standard) are not sequential and with a variable length. This means we cannot choose the same approach to manage the two data, but two different *sequential* and *for event*s modes, previously described, are necessary. This allows to the software developer to handle data those come from different devices during system operation without changing the class, decreasing the errors in case of different identification strings. This *Serial* class simplifies the implementation of UART protocols for different devices (RS232, RS485) and it allows all types of error checking, giving to the user the chance to create its own Cyclic Redundancy Check (CRC), as we did in our software solution.

The realized software is fully graphic and shows each operation on screen in order to provide to the common user or technician a clear vision of the current state of the PV plant, and therefore to allow quick intervention in case of failures. The figure 12 illustrates a polar diagram screenshot of the implemented software, which represents a projection from a hemispherical orbit to a circular two-dimensional representation; this screenshot of the Main Tab is a summary screen of daily solar track represented over a hemispherical graph, with all main information about the day, the time and the expected positioning of the sun. Consequently, this main tab is a user-friendly support to get a complete daily report of real-time data. Representation and data are both affected by distortion due to direct projection method; this distortion is so significant as much as the tracing is close to two-dimensional diagram edge, but it is not so much important for the correct tracking valuation of diagram, neither to user visual performance.

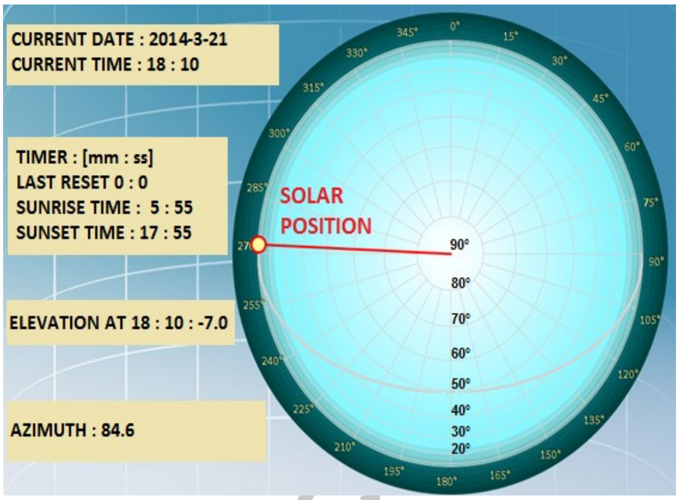


Figure 12. Typical screenshot of the implemented software, on PC with Master role, showing the solar position for driving the solar tracker motors

Fig. 13 shows a slave screenshot of the software with driven trackers (eleven totally) moved to right position from PC by cyclic addressing; this screenshot is a summary screen of the ongoing positioning transmitted to slave PV panels which is sent to one solar tracker at a time. In this Tab, on the right side, there are the daily position settings of pre-sunrise and post-sunset states for a fine calibration of these two daily events. On the left side, it is shown the custom string interface that can be sent by user's typing.

In addition to the normal addressing execution, into "Slaves tab", it is possible to send an individual data string for a particular tracker, if it is necessary to redefine an ongoing positioning. It is possible to choose the tracker from a list menu, but the command requires to define every single later label, respectively from the first position: slave tracker destination, master string (FF), direction East or West (2/1), most significant part of Azimuth byte, less significant part of Azimuth byte, direction Up/Down (2/1), most significant part of Elevation byte, less significant part of Elevation byte, checksum for any previously byte into string.

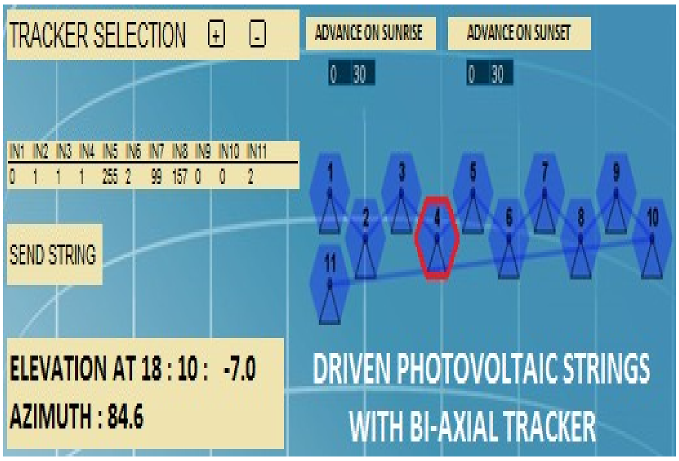


Figure 13. Slave screenshot of the implemented software, on PC with Master role, showing driven trackers (eleven totally) moved to right position from PC by cyclic addressing

Fig. 14 reports a graphic representation of the hemispherical azimuthal projection on the polar plan, produced by the designed application in order to improve the results comprehension of the user. This graphic helps the user to understand the unavoidable distortion due to projection and the representation of numerical data compression close to graph's edge.

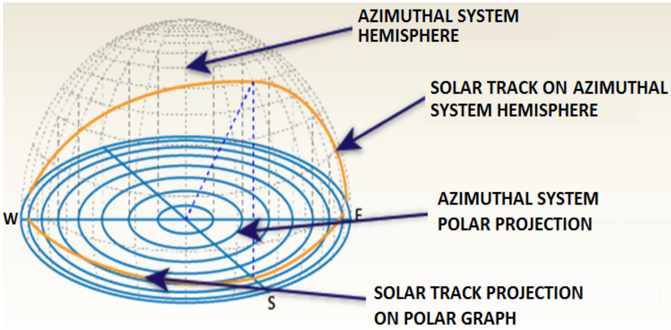


Figure 14. Graphic representation of the hemispherical azimuthal projection on the polar plan

In order to verify the accuracy of the algorithm for the calculation of the sun circadian orbit during a day from sunrise to sunset, the produced data related to azimuth position and elevation were compared with those provided on MeteoTitano website [9]. As shown in Fig. 15, the largest deviation is detected around the solar noon, while all other data can be superimposed; the found differences, however minimal, are considered negligible for the proper management of the solar trackers.

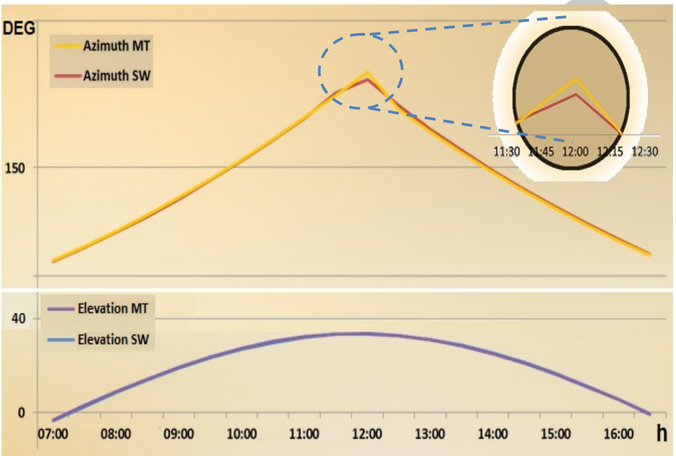


Figure 15. Graphic comparison between solar position data from Meteo Titano website and data obtained by software algorithm, from sunrise to sunset.

In conclusion, the realized software is able to determine the solar position through the algorithm and to communicate with the slave electronic boards, which drive the motors of the solar trackers, via PC [4] [10] [5]. The main improvements of the new PC-based equipment with respect to the previous electronic apparatus are:

- ✓ error checking on strings sent to the slaves.
- ✓ sunrise and sunset time scalability through graphic interface.
- ✓ angular coordinates of azimuth and elevation always available in database.
- ✓ visual report of any information on an user-friendly graphic interface.
- ✓ independence of the PC system from blackout condition.
- ✓ high accuracy in the calculation of the solar path.

In the hardware solution of solar tracker system, it isn't possible to modify the source of main software installed on electronic boards without being physically close to the hardware typically with a programmer. Instead the new software solution is installed locally on PC but it can be edited and ported to another OS or to another mobile device as long as there is a network connection. The current solution is stand-alone since it hasn't necessity to communicate with the user. In consideration of mobile devices evolution, it's important to have a program simply portable to different device, not only for control needs but above all for security needs [11]. The Processing possibility about safety and OS changing guarantees the software's longevity and few problems for all developers that in future will have to add new feature or bug fixes to source code; this considerations makes Processing one of the better cheaper solutions on the market.

The entire program is separated in threads, one for every library or specific function, that operate depending on the data present in their fields, initialized from the main cycle. Fig. 16 illustrates the table of one of those, the *PositionController* that is responsible for position managing of solar trackers.

class PositionController extends Thread	
Field	
+	int day, month, year, hour, minute, numOfSlaves, rideElevation, rideAzimuth, select;
+	String goingTo, newSessionStart, FixedPosition, SunrisePosition, ZeroPosition;
+	Float azimuthTemp, initialStateAzim, elevationTemp;
+	byte[] readByte;
+	int[] errors;
+	Boolean animate;
Method	
+	void start();
+	void run();
+	void quit();
-	int[] AddChecksum(int[] get);
-	int compareArray(byte[] first, byte[] second);
-	void testString(byte[] first, byte[] second, int[] In, int index);
-	void FixedDirect(int slave);
-	void SunriseDirect(int slave);
-	void UpDirect(int slave);
-	void DownDirect(int slave);
-	void EstDirect(int slave);
-	float[] coordinateTitano();
-	int inpulseElev();
-	int inpulseAzim();

Figure 16. Table of thread PositionController that shown the fields and methods of this class.

A thread is usually structured with the three main methods namely start, run and quit respectively to start, manage and stop itself inside the main cycle. In particular, in this thread, we have methods to add a checksum in queue of an array string as *AddChecksum*, to test the same one by comparison with*testString*, to calculate a particular direction about tilt and azimuth as *FixedDirect, SunriseDirect, UpDirect, DownDirect, EstDirect*. Moreover, other private methods such as *compareArray, coordinateTitano, impulseElev, impulseAzim*are used into previously commented methods.

At the beginning of the program, every Tab acquires the daily information included into year-long database. It was built previously from the secondary database software generator; this daily data are compared with ones of system calendar and if they are different the software provides to update them. Then, it runs the threads for solar tracker control and for RS485 network management, showing every graphic description inside any Tab. Now, if the RS485 port has been chosen into textbox settings, previously switched in Setup Tab, the settings are read and applied inside the software and otherwise the internal Timer (10 minutes long) is reset. If it is the first run, the initial position of every solar tracker is sent to every slave and the graphic is updated; otherwise every 10 minutes the current position of every solar tracker is updated and the Timer countdown is shown in Main Tab. At the end, if the user has selected a GPS port, the data coming from satellites are loaded and the system watch is updated; otherwise the program saves the data state and every graphic inside of every tab. From now on, it repeats itself from database updating statement.

VI. Experimental setup for testing of designed electronic equipment with PC-based software and slave driving boards

In Fig. 17 is shown the laboratory experimental setup for testing the electrical operation of the designed electronic equipment; the implemented firmware, with related PC application, sends control commands through the RS-485 port to the slave boards which drive the engines of solar trackers.

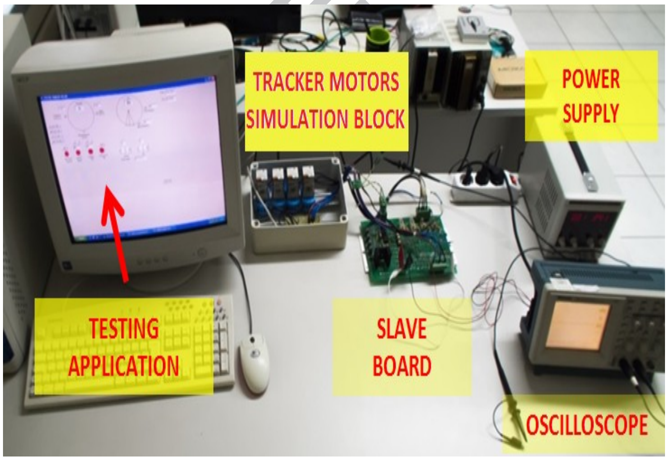


Figure 17. Experimental setup for system testing, with tracker motors simulation block, laboratory measurement instruments and implemented software running on PC.

The simulation block of tracker motors is connected to the PC's parallel port in order to reproduce the behavior of the solar tracker; four relays simulate the control of the two power motors (2 outputs for each motor) that change the value of the Azimuth and Elevation coordinates (Fig. 18). In the experimental setup shown in Fig.17, the software on PC reads the inputs from slave boards and sends output command strings to actuators from parallel port. Communication between RS232 serial port on PC and RS485 serial port on slave boards requires an adaptation block in order to convert the voltage values used to represent the different logical levels; finally, an opto-coupler simulates the behavior of each proximity sensor.

Figure 18. Zoom on the 4 actuation relays for motor control in the simulation block of the tracker motors.

Fig. 19 shows a PC screenshot of the Master software for monitoring the proper position of solar trackers during normal photovoltaic plant activity; the picture shows the position reached by tracker at the coordinates set immediately after the opening of the program and the database data loading. This screenshot of the solar tracker emulator executed on a secondary PC simulates the activities of a biaxial solar tracker, showing status of azimuth and tilt limit switches and the counting of the pulses to reach them. The hemispherical graph is represented by two polar graphs which show every new value about azimuth and tilt position, received from the Main program and executed on main PC connected by RS485; it is updated after every transmission inside the graph's textboxes as shown on the upper side. The picture is completed by a list of position reset buttons on the left side. This software has been useful to monitoring the proper position of solar trackers during activity tests of PV plants.

Finally, Fig. 20 illustrates two pictures of the oscilloscope's screen captured during the testing of the designed equipment; in the left image, it is shown the EST command signal sent through the software and running to the slave board on the RS485 communication line, while in the right image, it can be seen, together with the previously launched command, the response of the slave board recognized by different amplitude of the signal level.

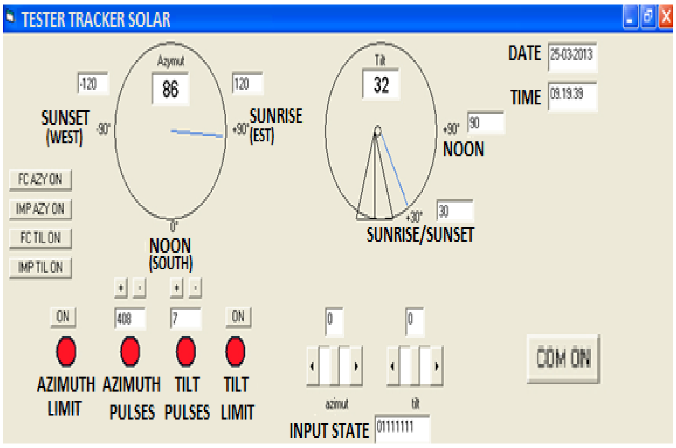


Figure 19. PC screenshot of Master software for monitoring the proper position of solar trackers during normal photovoltaic plant activity

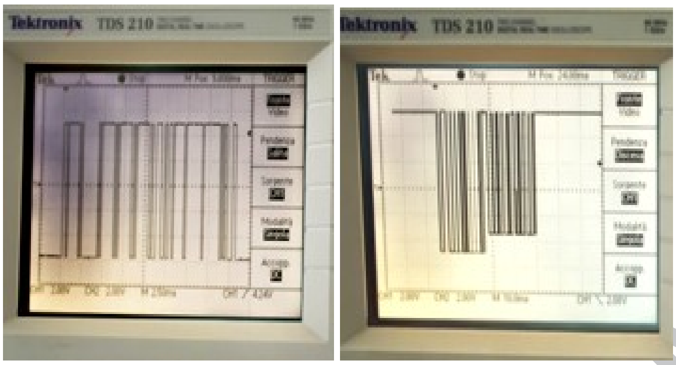


Figure 20. Oscilloscope detection of command signals from Master software to slave driving boards (left) and response signals from same slave boards (right).

Figure 21. Photovoltaic plant with bi-axial solar trackers for final testing of the realized electronic equipment

VII. CONCLUSION

Solar trackers continue to be an optimal solution for increasing the efficiency of a PV plant and thus its energy production even if an economic comparison must be made with respect to fixed plants since the cost of electric drivers must be taken into account [11] [12]. With regard to cost's amortization of once installed control/driving systems, it is determined by few essential factors: the cost of general control electrical panel (about 1.000 $), the cost for each solar tracking device (about 300 $), the government incentives eventually granted and the price of the energy, produced by the photovoltaic plant, paid from the grid operator to which the PV plant is connected. Just to give you an example, for a 1 MW PV plant with one hundred solar trackers, the investment will be 1000 $ + (300 $ * 100) = 31.000 $. The 1MW plant's energy production (if installed in the South of Italy), without solar trackers, will be 1.400.000 kWh/year; with installation of biaxial solar trackers, instead, the additional production of energy for each year will be 1.400.000 kWh * 0,35 = 490.000 kWh supposing an energy production gain of about 35%. With an incentive of 18 cent/kWh, we will have an income of 490.000*0,18=88.200 $/year whereas from sale of 490.000 kWh produced energy at 3 cent for each kWh, we will obtain other 14.700 $. From obtained data, the amortization time varies from a minimum of about 4 months to a maximum of 2.1 years without government incentive.

As a dynamic system, subject to changing condition in the operating mode, a PV plant based on tracking system must be reliable and the PV modules, used for this purpose, must be also viable for reaching the objectives [13]. This kind of plant has a high cost of managing if located in dusty and industrial area since it is necessary to provide a continuous cleaning of PV modules in order to prevent the decreasing of energy production [14] [15] [16]. However, with the proposed algorithm and related software running on PC, we have demonstrated that a further increase of the efficiency is still possible using any PV technology [17][18][19]. The designed and realized electronic system is able to control and drive bi-axial solar trackers, in order to maintain the panel's surface always perpendicular to solar rays, by means of a PC software application which allows user-friendly graphical interface and simple and rapid PV plant management.

Cavalera company, in collaboration with the University of Salento, installed several designed control/driving systems in both old fully-hardware solution and, after, with new software able to control/drive by PC the slave solar trackers. In order to verify the system's reliability, long-term tests have been made, checking after few and several months the correct operation of installed systems and the proper movement of PV strings following solar orbit during the day, without significant positioning errors.

REFERENCES

[1] A. Lay Ekuakille, G. Vendramin, A. Fedele, L. Vasanelli, A. Trotta, "PV Maximum Power Point Tracking Through Pyranometric Sensor: Modelling and Characterization", International Journal On Smart Sensing And Intelligent Systems , Vol.1- n.3, pp.659-678, (2008).

[2] R. Gregor, Y. Takase, J. Rodas, L. Carreras, A. Lopez, M. Rivera "A Novel Design and Automation of a Biaxial Solar Tracking System for PV Power Applications", IEEE Proc. of 39 th Conference of Industrial Electronics Society (IECON 2013) , pp. 1484-1489 (2013).

[3] M. Bortolini, M. Gamberi, A. Graziani, M. Manfroni, R. Manzini, "Hybrid Strategy for Bi-Axial Solar Tracking System",Journal of Control Engineering and Technology, Vol. 2 Issue 4, pp 130-142 (2012).

[4] S. Zubair, A. Suleiman, H.T. Abdulazzez, B.A. Salihu, "Design And Construction of a Microcontroller Based Single Axis Solar Tracker", Innovations in Science and Engineering 1, pp 41-47 (2011).

[5] M.T.A. Khan, S.M.S Tanzil, R. Rahman, S.M.S Alam, "Design and construction of an automatic solar tracking system", Proc. of IEEE Int. Conf. of Electrical and Computer Engineering, pp326-329 (2010).

[6] P. Visconti, V. Ventura, F. Tempesta, D. Romanello, G. Cavalera, "Electronic system for improvement of solar plant efficiency by optimized algorithm implemented in biaxial solar tracker", Proceeding of 10thIEEE EEEIC 2011,ISBN 978-83-932625-0-2, pp 434-437.

[7] A. Lay-Ekuakille, G. Vendramin, A. Fedele, L. Vasanelli, D. Laforgia, A. Trotta "Kriging Code for the Treatment of Environmental and Electrical Parameters for Photovoltaic Panels",Proc. of 6th IEEE International Multi-Conference on SSD (SSD'09), Djerba Tunisia (2009).

[8] S. Ranjita Nayak, C. Ranjan Pradhan, "Solar Tracking Application", IOSR Journal of Engineering, Vol. 2(6), pp 1278-1281 (2012).

[9] www.meteotitano.net

[10] A. Kassem, M. Hamad, "A microcontroller-based Multifunction solar tracking system", Proceeding of 2011 IEEE InternationalSystems Conference, ISBN 9781424494941, pp 13-16 (2011).

[11] W. Wei, L. Shaoyuan, "Model predictive control of 2axis solar tracker for solar energy system", Proc. of IEEE 31th Control Conference, pp 4177-4182, ISBN 9781467325813 (2012).

[12] A. Ruelas, N. Velázquez, L. González, C. Villa-Angulo, O. García, "Design, Implementation and Evaluation of a Solar Tracking System Based on a Video Processing Sensor", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 10, pp172-178 (2013).

[13] C. Calo', A. Lay-Ekuakille, P. Vergallo, C. Chiffi, A. Trotta, A. Fasanella, A.M. Fasanella, Measurements and Characterization of Photovoltaic Modules for Tolerance Verification, IJMTIE, International Journal of Measurement Technologies and Instrumentation Engineering, Vol.1 n.2, pp. 73- 83, April-July 2011.

[14] P. Vergallo, A. Lay-Ekuakille, A. Ciaccioli, G. Griffo, "Compensating Environmental Influencing Factors for Characterizing PV module Efficiency", 5th Imeko TC19 "Environnmental Measurements", 23-24 september 2014, Chemnitz, Germany.

[15] A. Lay-Ekuakille, A. Arnesano, P.Vergallo, (2013) "Thin-film-based CdTe PV Module Characterization: Measurements and Energy Prediction Improvement", Review of Scientific Instruments, Vol.84, pp.15114-7 (2013).

[16] R. Sharma, G. Singh, M. Kaur "Development of FPGA-based Dual Axis Solar Tracking System", American Transactions on Engineering and Apllied Sciences,Volume 2 - No. 4, pp253-267 (2013).

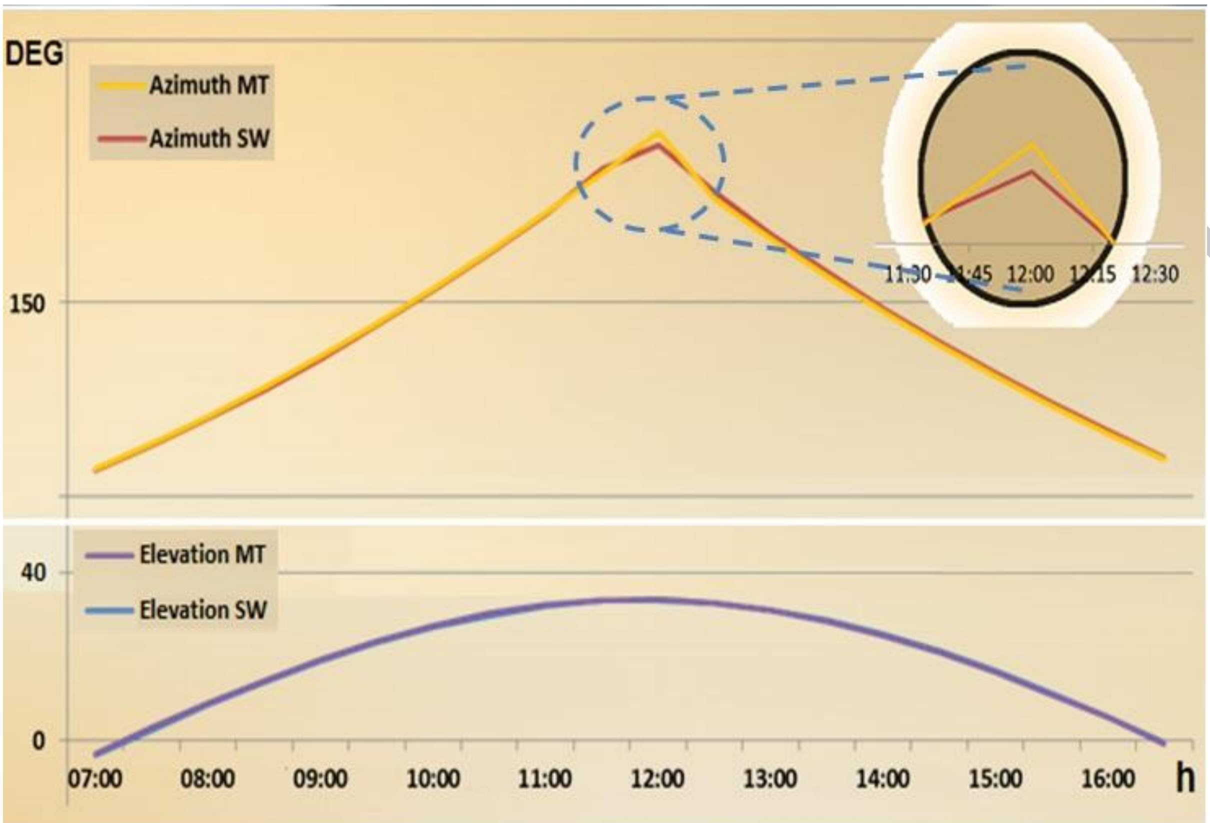
[17] B. Krishna, K. Sinha,"Tracking of sun for solar panels and real time monitoring using LABview", Journal of Automation and Control Engineering Vol. 1, No. 4, pp 312-315 (2013).

[18] H. Mousazadeh, A. Keyhani, A. Javadi, H. Mobli, K. Abrinia, A. Sharifi, "A review of principle and sun-tracking methods for maximizing solar systems output" , Renewable and Sustainable Energy Reviews – Elsevier, Vol. 13, pp. 1800–1818, (2009).

[19] S. Bazyari, R. Keypour, S. Farhangi, A. Ghaedi, K. Bazyari, " A Study on the Effects of Solar Tracking Systems on the Performance of Photovoltaic Power Plants" Journal of Power and Energy Engineering, Vol. 2, pp. 718-728, (2014).

Highlights

- System for controlling/driving solar trackers by means of a PC software application
- PV panels driven by designed setup follow solar orbit for higher energy production
- By PC application, user can manage, control and reset system parameters
- New findings on tracking system measurements have been displayed



DEG
Azimuth MT
Azimuth SW
150
11:30
11:45
12:00
12:15
12:30
Elevation MT
Elevation SW
40
0
07:00
08:00
09:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
h