A STUDY OF PHOTOVOLTAIC ARRAYS UNDER DIVERSE OPERATING CONDITIONS

By

DAVID CHRISTOPHER SCHOOLEY

Bachelor of Science
Oklahoma State University
Stillwater, Oklahoma
1987

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
July, 1989

A STUDY OF PHOTOVOLTAIC ARRAYS

UNDER DIVERSE OPERATING CONDITIONS

Thesis Approved:

R. Ramakumar

Thesis Advisor

Richard L. Cummins

H J Allison

Norman N. Durham

Dean of the Graduate College

PREFACE

This thesis is the result of a study into the operating characteristics of photovoltaic arrays under diverse operating conditions. Computer simulations of PV arrays were performed using a range of insolation conditions and cell fill factors. Conclusions were then reached for several different array configurations as follows:

1. Tracking arrays were compared to fixed position arrays.
2. The performance of loads designed to maximize energy utilization efficiency were compared to the performance of loads designed to maximize power output.

There are many people who deserve part of the credit for the success of this study. First, I would like to thank my advisor, Dr. R. G. Ramakumar for his guidance and everlasting patience, and for providing financial support through the Engineering Energy Laboratory at Oklahoma State University and its sponsoring utilities during work on this thesis. The other two members of my committee, Dr. H. Jack Allison and Dr. Richard L. Cummins also deserve a special thanks for their time and effort.

In addition, I would like to thank the rest of the faculty and staff of the School of Electrical and Computer Engineering at O.S.U. for all of the fun and friendship over the past several years.

Finally, I would like to thank my parents, Neal and Shirley Schooley for their love and encouragement while I was in school at O.S.U., and especially while I was trying to get this thing finished.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

A	=	Cloud area in square nautical miles
a1,a2,a3	=	Constants used in distribution function F(A)
e	=	Electronic charge, 1.602E-19 coulombs
$E_{BASE,F}$	=	Base value for energy (Fixed arrays)
$E_{BASE,T}$	=	Base value for energy (Tracking arrays)
E_N	=	Normalized output energy
E_{NMAX}	=	Maximum obtainable output energy (Normalized)
$F(A)$	=	Cumulative probability distribution function used in generating clouds.
I_{BASE}	=	Base value for current
I_L	=	Load current
I_N	=	Normalized load current
I_O	=	Normalized dark current
I_S	=	Normalized short-circuit current
k	=	Boltzmann's constant, 8.6168E-5 eV/°K
MPPT	=	Maximum Power Point Tracker
η_E	=	Energy Utilization Efficiency
P	=	Power
P_{SUN}	=	Ratio of peak clear-sky insolation to maximum possible over array
$\%_{CLOUD}$	=	Percent of sky covered by clouds

NOMENCLATURE (Continued)

$\%_{SUN}$	= Insolation over the array as a percent of the base value of clear-sky insolation
$\%_{TR}$	= Transmissivity in percent
P_{TR}	= Fraction of sunlight transmitted through clouds
R_{BASE}	= Base value for resistance
R_{PU}	= General normalized load resistance
$R_{PU,MP}$	= Normalized load resistance optimized for maximum power output
$R_{PU,\eta}$	= Normalized load resistance optimized for maximum utilization efficiency
$t_{SUN,i}$	= The number of equally spaced data points during time segment i during which the sky above the array is clear
$t_{CLOUD,i}$	= The number of equally spaced data points during time segment i during which a cloud is present above the array
T	= Temperature, °K
T_{SUN}	= The total number of equally spaced data points during which the sky above the array is clear
T_{CLOUD}	= The total number of equally spaced data points during which clouds are present above the array
t_i	= The number of equally spaced data points making up segment i
T_{TOT}	= The total number of equally spaced data points making up one day
V_L	= Load Voltage
V_N	= Normalized load voltage
V_{OC}	= Open circuit voltage

V_{BASE} = Base value for voltage

CHAPTER I

INTRODUCTION

The objective of this study is to examine the performance of photovoltaic (PV) arrays under various operating conditions for single-day periods. Given that photovoltaic arrays are designed so that long-term performance is optimal, how is performance affected when things are not optimal on a day-by-day basis? For instance, the array may be optimized for an average insolation level of 0.9 kW/m^2, but on any given day the average insolation might be higher or lower than this and the instantaneous insolation also varies during the day. If the array is designed for a certain level of cloudiness, how will the array perform when it is cloudier than normal, or if the day is clear? Also, how does the performance differ for cells of various fill factors? These issues will be studied using computer simulations of a typical PV array.

Introduction to Solar Cells

With growing concerns over global warming due to the continued use fossil fuels, new energy sources are needed which are less hostile to the environment. The use of solar energy technologies, particularly photovoltaics, can serve this need. The cost of PV cells is still too high for widespread use of the technology, although researchers are hopeful that recent advances can bring the costs down and make them economically competitive with more traditional methods of generating electricity.

There are three major types of PV cells: single-crystal, polycrystalline, and thin-film. The single-crystal cells, which are made from wafers of semiconductor consisting of a single crystal, are the most efficient type of cell, but are the most costly. The thin-film cells are the least efficient, but are the cheapest.[1] Table 1 gives a comparison of the best PV cell efficiencies that have been obtained in laboratories to date. In addition to listing the efficiencies according to cell type, Table 1 also gives efficiencies for when the incident solar radiation is concentrated to values much higher than the normal level of insolation.

TABLE 1

PV-CELL EFFICIENCIES

Technology	Best efficiency
Single Crystal	
Gallium arsenide/aluminum gallium arsenide	27.6
Gallium arsenide	24.8
Single-crystal silicon wafer	22.3
Polycrystalline	
Cast silicon wafer	17.8
Dendritic-web silicon	16.7
Thin silicon	15.7
Edge-defined film-fed growth (EFG) silicon	14.5
Thin film	
Copper indium diselenide	14.1
Amorphous silicon (multijunction)	13.3
Cadmium telluride	12.3
Concentrators (degree of concentration)	
Gallium arsenide (206x)	29.2
Silicon (140x)	28.2

Source: Glenn Zorpette, "Photovoltaics: technical gains and an uncertain market", *IEEE Spectrum*, Vol. 26, No. 7, pp. 42-43, July 1989

A set of current-voltage (I-V) characteristics for a solar cell is shown in Figure 1. In addition to the cell characteristics, the load-line for a load resistance is also shown. The operating point of the cell for any value of insolation may be found where the load-line intersects the cell I-V curve corresponding to the desired level of insolation. The dashed line in Figure 1. passes through the locus of

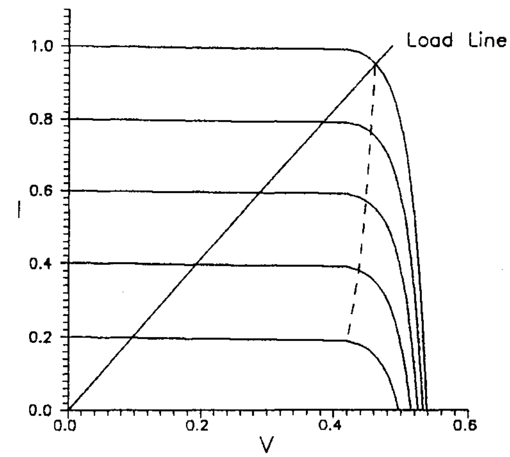

Figure 1. I-V Curves for a Solar Cell

maximum power points of cell.

The maximum power points of a solar cell are those points on the I-V curve where the output power is at a maximum. By designing the solar array so that the load-line passes through these points, the output power of the array can be maximized. The insolation on the array is rarely constant however, and it can be seen from Figure 1. that for most insolation values, the operating point of the array will not be at one of the maximum power points.

When studying the performance of loads supplied by solar cells, one must take into account the fact that the solar cell is neither a perfect voltage source nor a perfect current source. When the operating voltage of the load is low, the solar cell behaves as a current source. When the operating voltage of the load is high, the cell more closely approximates a perfect voltage source. Complicating the issue is the fact that to obtain the maximum available power from the cell, the operating point must lie in the neighborhood of the knee of the curve. Also, the operating point of the cell depends on the incident solar radiation and cell temperature. The change in output current of the cell is linearly related to the change in insolation and the change in output voltage is a linear function of the change in temperature.

Introduction to Insolation

The mathematical models used to study insolation (incident solar radiation) usually consist of two parts, a deterministic or clear-sky component and random component. The parameters for the deterministic part depend on such factors as the time of year, the location of the array on the globe, and the mechanical design of the array. The random part is due to clouds, which block out part of the clear-sky insolation.

When insolation is studied for the purposes of determining PV array performance, the configuration of the array must also be taken into account. There are several possible configurations for mounting solar arrays. The array may track the sun as it moves across the sky, or the array may be stationary and point solely in one direction. Tracking arrays are able to supply more energy during the day, but cost more and are more difficult to maintain. If an array tracks the sun, it may do so either on two axis of rotation or just one. If the array tracks on a single axis, or is completely fixed, it is usually tilted at an angle equal to the degree of latitude in order to maximize the incident insolation[2].

The incoming sunlight may be concentrated in order to increase the solar cell efficiency and increase the output per unit area. The factor by which the sunlight is concentrated in practice ranges from approximately 20 to 1000[3].

CHAPTER II

LITERATURE REVIEW

Array Performance

Most of the work involved with studying the characteristics of loads powered by solar arrays has involved long-term studies to determine yearly efficiencies. A few researchers however have studied the operation of PV arrays on a daily basis.

A paper by J. Appelbaum[4] discusses the performance of loads under several operating configurations. Two types of loads are considered: pure resistance and water electrolyzer ($V_L = V_0 + IR_P$), with the four operating configurations as follows:

1. A single load is powered by a single PV panel.
2. Parallel combination of two loads powered by a common source of two solar arrays.
3. A single load and solar array with a Maximum Power Point Tracker (MPPT).
4. Two loads powered by a common source of two solar arrays with a MPPT.

The two circuit configurations are shown in Figure 2.

Appelbaum uses the concept of energy utilization efficiency to measure

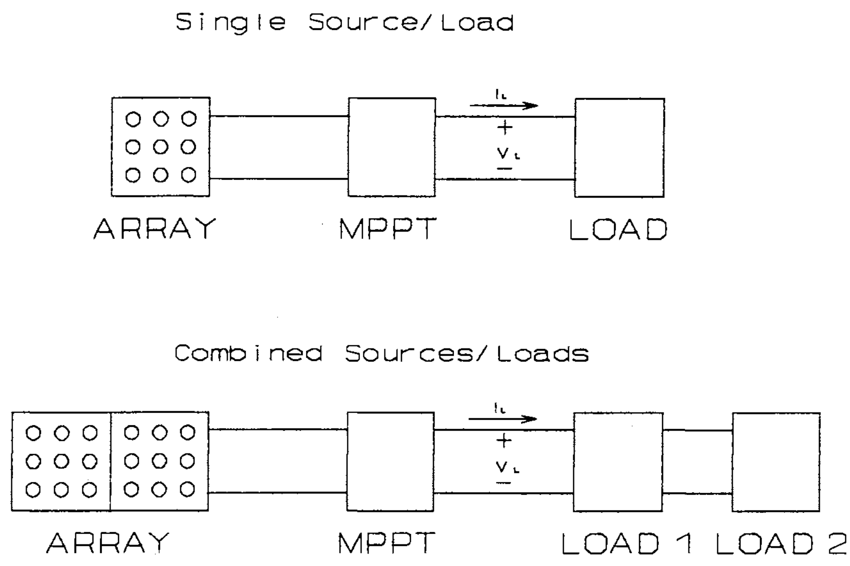

Figure 2. Source/Load Combinations Studied by Appelbaum

the performance of the above configurations. The energy utilization efficiency of a solar array is defined by:

$$\eta_E = E_N/E_{NMAX}; \tag{1}$$

where E_N is the normalized energy supplied by the array during the entire day, and E_{NMAX} is the normalized energy that would be supplied if the load was exactly matched to the solar cell characteristics at all times. Since the load cannot be exactly matched to the array at all times, the utilization efficiency will always be less than 1.0 unless an ideal MPPT is used.

Two of the conclusions reached by Appelbaum are:

1. When the performance of a system with two loads and a common source is compared to that of a system where each load is powered individually, the performance of one load will improve, and the performance of the other will worsen.

2. Although the performance of one load decreases when the two loads are connected to a common source, the net performance of the two loads increases.

The loads used in the study were not matched to the maximum power points of the arrays, and the arrays which were connected in parallel with each other were identical to the arrays connected singly.

The results here may seem strange at first because of where the load-lines lie in relation to the maximum power points. It will be demonstrated later in this paper that the operating points which maximize η_E lie to the right of the maximum power points. The load-line of the load which has its performance decreased lies to the left of the maximum power points, and the load-line for the other load lies to the right. When the loads are combined, while the resistance of the loads in parallel is to the left of the maximum power points, the resistance that is seen by each source lies to the right and near to the points which maximize η_E, which leads to the increase in performance.

C. C. Gonzalez and co-workers at Jet Propulsion Laboratory[5,6] have included the effects of fill factor and temperature in a large-scale study of PV array performance. Cell efficiency was studied here, but the annual energy output was studied instead of the daily output. One of the primary purposes of the study was to determine if the performance advantages of maximum power tracking over fixed voltage tracking are worth the extra cost. One problem with fixed voltage operation is that the optimum operating voltage decreases as the array ages (the fill factor decreases.)[6] In modeling the sensitivity of voltage tracking to changes in fill factor, fill factors ranging from 0.45 to 0.75 were incorporated into the analysis. To investigate temperature effects, the nominal operating cell temperature (NOCT) for each cell in the study was used instead of the more commonly used 300°K and $1kW/m^2$ standard operating conditions (SOC). Values of NOCT used ranged from 318°K to 343°K, depending on the array mounting configuration. One of the primary conclusions reached by Gonzalez, et al, was that even without considering losses by a real MPPT, MPP tracking is only economically worthwhile if the additional cost over fixed-voltage tracking is less than 2% of the value of the array's annual energy output. This is relevant to the study at hand because in some situations intended use of the array may not justify the additional cost of MPPT's.

Khallat and Rahman[7] have developed a probablistic method to predict PV performance using long term climatological data. For each season, a set of probability distribution functions are fit to the insolation data on an hourly basis. The insolation data is bimodal, so it is divided into segments, and a unimodal distribution is then fit to the data. Table 2 gives an example.

TABLE 2

PROBABILITY DENSITY FUNCTIONS FOR SPRING

Time	Segment 1	Segment 2
8 a.m.	Beta	Beta
12 p.m.	Weibull	Beta
2 p.m.	Beta	Beta
5 p.m.	Beta	Weibull

Source: M. A. Khallat and Saifur Rahman, "A Probabilistic Approach to Photovoltaic Generator Performance Prediction", *IEEE Transactions on Energy Conversion*, Vol. EC-1, No.3, pp. 34-40, September, 1986

After finding the correct probably distribution, the expected power output of the array can be found using the cell parameters and the expected insolation. A "capacity factor" is then presented which is identical to the energy utilization efficiency as given by Appelbaum. The method suggested by the above paper is useful for determining how well the array performs on average, but does not explicitly take cloudiness into account.

Insolation

The data used to develop insolation models are often collected on an hourly basis, although data collected more often lead to better models. Measured insolation is highly dependent on location where the data are collected and the resulting models are usually statistical in nature[8]. The insolation model used in this thesis will be more general than those presented in the following papers.

J. E. Sherry and C. G. Justus[9] have studied the effects of clouds on the basis of cloud observations taken every three hours. The observations were used to develop models for several different cloud types.

Chowdhury and Rahman[10] have modeled solar irradiance as a combination of a clear sky component (deterministic) and a cloud cover component (stochastic). The model used to predict the cloud component is an Auto-Regressive Integrated Moving Average (ARIMA) model which was found to be reasonably accurate in predicting PV output.

A fairly accurate model for modeling insolation on clear days was developed by El-Adawi and fellow researchers in Saudi Arabia.[11] The intensity of radiation received on a flat plate was recorded and averaged at separate locations on clear days in March and April. This paper also includes an equation for clear-sky insolation in terms of the solar constant, direct and solar flux, and the solar position. The use of this equation would allow the inclusion of cloud formations in the model, if the changes in direct and solar flux were known for the desired cloud type.

CHAPTER III

INSOLATION MODELING

General Modeling

To model the insolation during a day, a data file is generated with insolation values for the day in question. Since two types of arrays are modeled (tracking and fixed arrays), two different mathematical functions are used to generate the clear sky insolation. A sine function is used when simulated fixed arrays, and the clear-sky insolation is assumed to be constant when simulating tracking arrays. By multiplying the proper function by a given maximum value of normalized insolation (P_{SUN}), the performance for a clear day in August, for instance, can be compared to a clear day in March. Both the length of the day and the time interval between values can be varied during the simulation.

Fixed Arrays

For non-tracking arrays, a cloudless day can be modeled using the positive half of a sine function as shown in Figure 3. The parameters necessary for modeling clouds are the cloud transmissivity, size, speed, and the percentage of the sky covered by clouds. To model the effectS of clouds, the clear sky insolation is multiplied by the cloud transmissivity if a cloud is present during that time period.

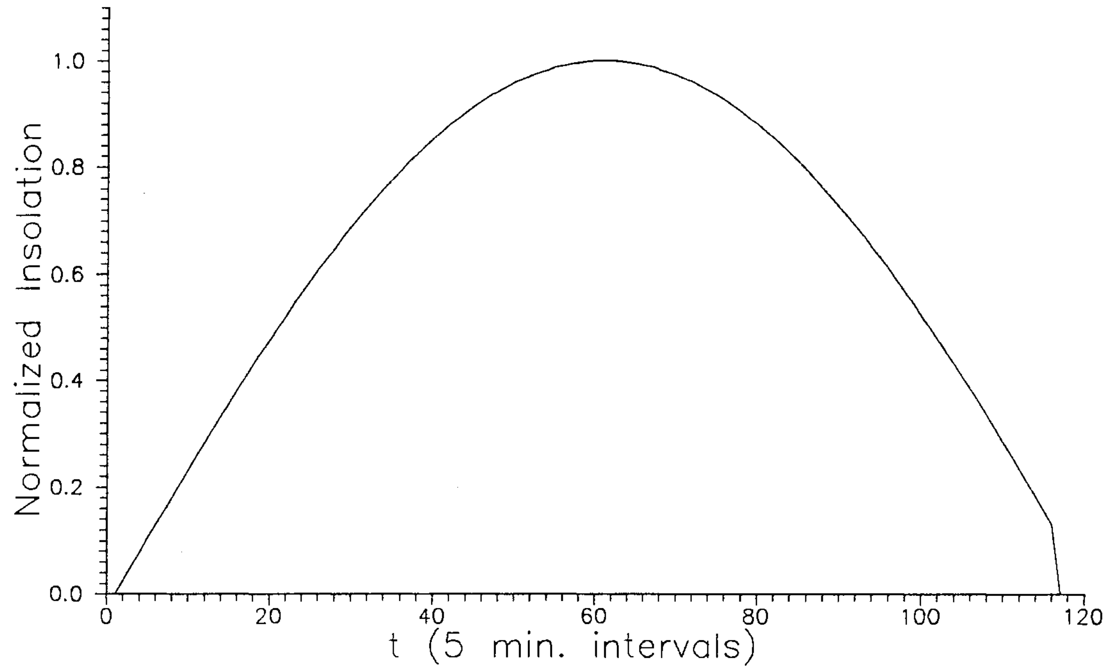

Figure 3. Insolation for Fixed Array, Clear Day

To determine if a cloud is present, a random number between zero and one is used as an input to an iterative equation which determines cloud area. After the cloud area is determined, its square root is taken to determine cloud length. The photovoltaic array is considered to be a point relative to the size of the cloud, and therefore no distinction is made between cloud length and width. After the cloud length is found, the length of time ($t_{CLOUD,i}$) that the cloud is over the array is found using the cloud speed. The clear sky insolation is then multiplied by the transmissivity for that length of time, and sky over the array is then assumed to be clear again for a time $t_{SUN,i}$, given by

$$t_{SUN,i} = t_{CLOUD,i} \times [(1/P_{CLOUD}) - 1]; \tag{2}$$

where P_{CLOUD} is the fraction of sky filled with clouds.

To derive the above equation, consider a day consisting of T_{TOT} data points spaced equally in time. If the percent of sky covered by clouds is equal to $\%_{CLOUD}$, then $P_{CLOUD} = \%_{CLOUD}/100$ and the number of data points with clouds over the array is:

$$T_{CLOUD} = T_{TOT} \times P_{CLOUD} \tag{3}$$

T_{TOT} is given by:

$$T_{TOT} = T_{SUN} + T_{CLOUD} \tag{4}$$

Solving for T_{TOT} in equation (3) and inserting into equation (4) gives:

$$T_{CLOUD}/P_{CLOUD} = T_{SUN} + T_{CLOUD} \tag{5}$$

so:

$$T_{SUN} = T_{CLOUD} \times [(1/P_{CLOUD}) - 1] \tag{6}$$

Now consider a day to be divided into N segments of variable lengths t_i, with each segment consisting of a stretch of cloudy data points followed by a stretch of clear-sky data with

$$t_i = t_{SUN,i} + t_{CLOUD} \tag{7}$$

The relationship between $t_{SUN,i}$ and $t_{CLOUD,i}$ is assumed to be similar to that between T_{SUN} and T_{CLOUD}. Since:

$$T_{TOT} = \sum_{i=1}^{N} t_i = \sum_{i=1}^{N} (t_{SUN,i} + t_{CLOUD,i}) \tag{8}$$

then

$$T_{CLOUD} \underset{i=1}{\overset{N}{=}} \Sigma\ t_{CLOUD,i}, \tag{9}$$

which means that by generating N data segments, each made up of a set of cloudy data and a set of clear data, we can generate a day with the correct percentage of cloud cover. The use of segments with variable time duration allows the cloud size to vary. If the segment durations were all equal, it would not be possible to vary the length of time a cloud is present, and at the same time generate days with a desired percentage of cloud cover since each segment starts at the beginning of a cloudy interval and ends at the end of the immediately following clear interval.

An example of the results for a fixed array on a cloudy day is shown in Figure 4. The figure shown is for a day with $P_{SUN} = 1.0$ and $P_{CLOUD} = 0.25$, with large sized clouds at a speed of 10 mi/hr. A large cloud is considered to have an area between 0.100 and 10.163 square nautical miles.[12]

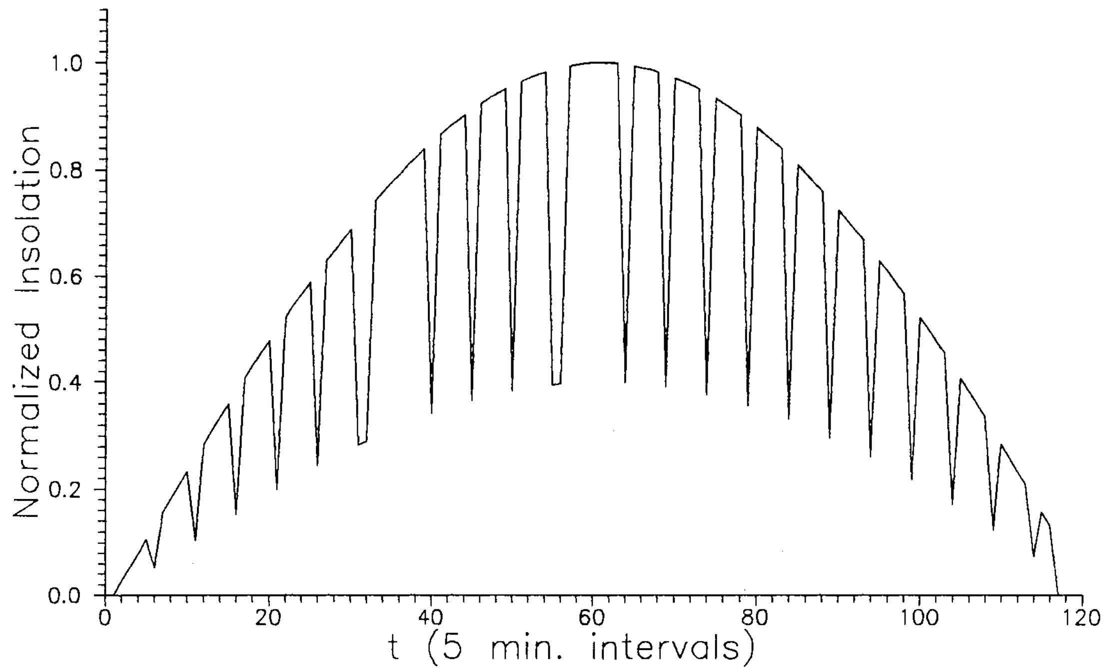

Figure 4. Insolation for Fixed Array, Cloudy Day

Tracking Arrays

For tracking arrays, the insolation for a cloudless day can be assumed to be nearly constant for the length of the day, and is set equal to P_{SUN} (Figure 5.). Thus, for the purposes of the simulations here, tracking arrays will be modeled using a constant clear sky insolation.

The movement of clouds over the array is modeled multiplying P_{SUN} by the cloud transmissivity over an appropriate interval of time (Figure 6.). Because the insolation is constant for a tracking system, the clouds are not simulated individually

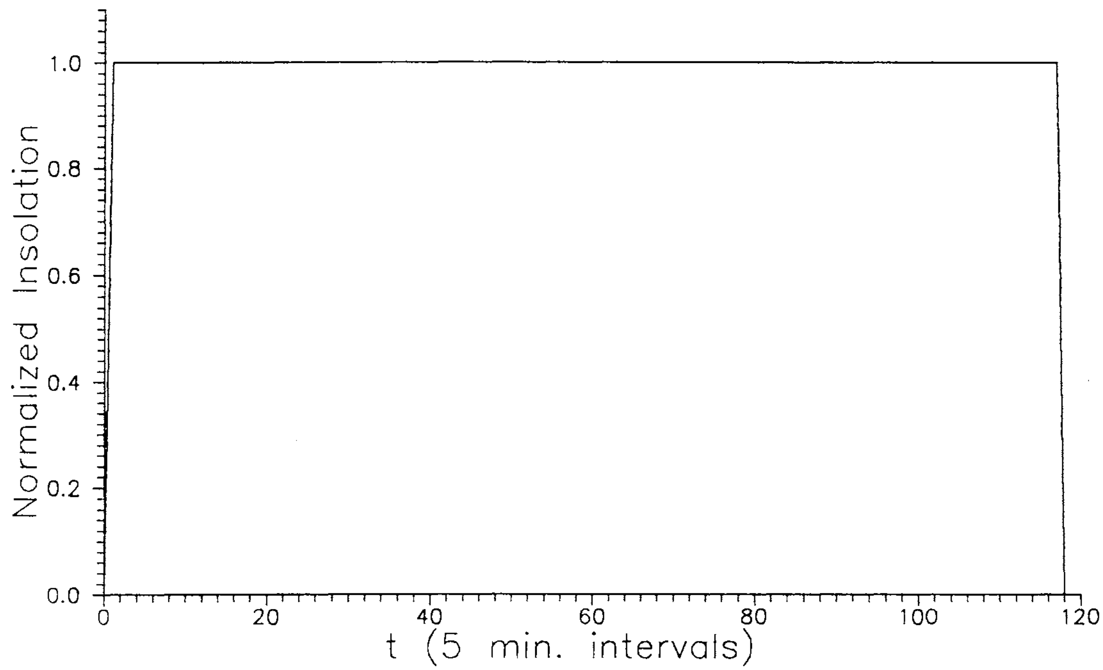

Figure 5. Insolation for Tracking Array, Clear Day

as when simulating fixed arrays, but instead the clear sky insolation is multiplied by the cloud transmissivity for a length of time equal to T_{CLOUD} and the sky is considered to remain clear for the remainder of the day (T_{SUN}).

Software Implementation

In the simulation program, the procedure responsible for generating insolation data is labeled 'DayGen'. The user is first prompted for the type of insolation pattern to be generated, that for tracking arrays or fixed arrays. Fixed arrays then require the desired cloud size and speed to be entered. Finally, the user is

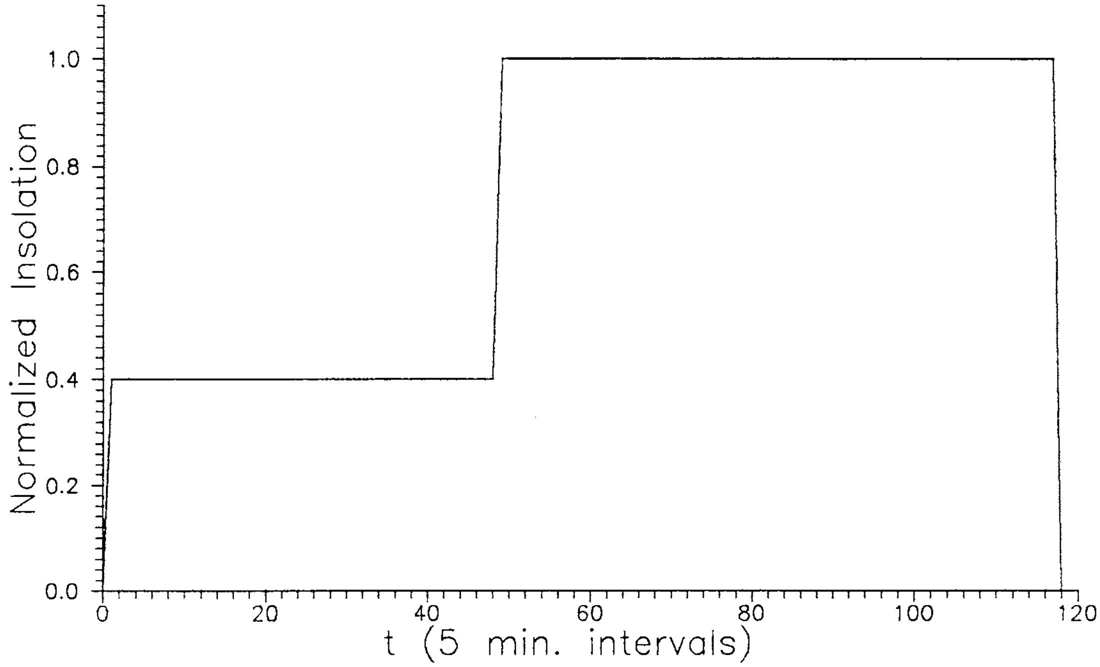

Figure 6. Insolation for Tracking Array, Cloudy Day

prompted for the $\%_{TR}$, $\%_{Cover}$, and $\%_{SUN}$ parameters. Any one or all of these parameters may be entered as a range of values by entering initial and final values along with an increment.

The equation used to generate cloud size in square nautical miles over fixed arrays is as follows[12]:

$$F(A) = A \qquad [a1 - a2 \ln(A)] \qquad - \qquad a3 \tag{10}$$

The constants a1, a2, and a3 depend on the size of the clouds being simulated and are given in Table 3.

TABLE 3

CONSTANTS USED TO DETERMINE CLOUD SIZE

Cloud Size (square nautical miles)	a1	a2	a3
0.1 < A < 2.690	0.880	0.442	0.180
0.1 < A < 10.163	0.346	0.104	0.059
0.1 < A < 30.884	0.147	0.033	0.022

Source: Ward T. Jewell, *The Effects of Moving Cloud Shadows on Electric Utilities with Dispersed Solar Photovoltaic Generation*, Ph.D. Dissertation, Oklahoma State University, 1988

CHAPTER IV

CIRCUIT ANALYSIS

Base Values for Circuit Parameters

All of the values for circuit parameters and insolation used in the simulations are normalized with respect to corresponding base values. The use of normalized values makes the simulation results more applicable to arrays of different sizes.

When simulations are done using actual insolation data, the values are converted to normalized values between zero and one by dividing each data point by a reference value equal to the maximum possible insolation. During the initial phase of this study, insolation data from monitoring equipment located on the roof of Engineering South on the Oklahoma State University campus in Stillwater, OK, were used in the simulations. For the purposes of this study, the reference value for insolation is assumed to be equal to $1kW/m^2$[13]. When test days are generated by the computer, all data points are simply given initial values between zero and one. Once the normalized insolation is found, the normalized short-circuit current is known, and the load current and load voltage can be found by using equation (11) as the governing equation in the circuit analysis.

The circuit used in the analysis is shown in Figure 7., with I_N and V_N as the normalized PV output current and voltage, respectively. The base value for current

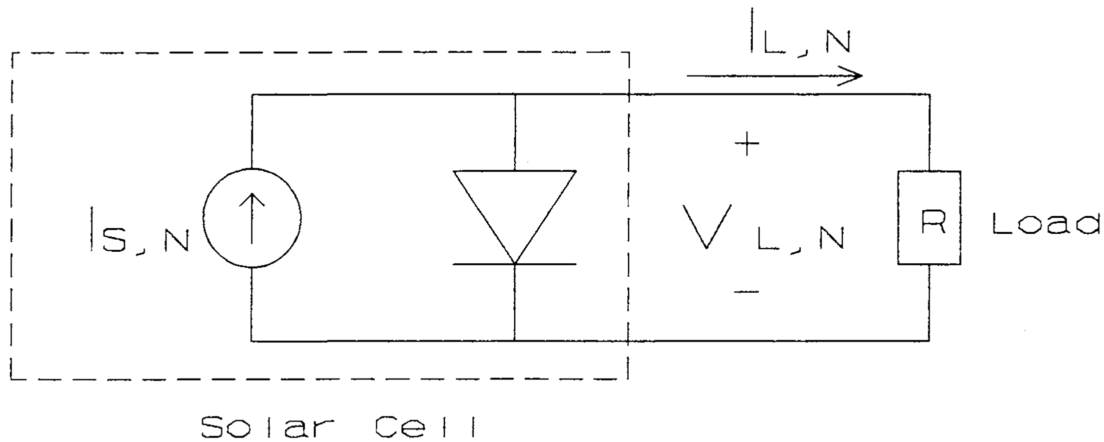

Figure 7. Test Circuit

is assumed to be 1.0 A, occurring at an insolation level of 1kW/m². The base value for voltage (V_{BASE}) is taken as the open-circuit voltage of a reference cell with I_O = 1.0E-13 p.u. at a temperature of 300 °K, and with I_S = 1.0 p.u. The fill-factor for this pseudo-ideal cell works out to be about 0.857. Since the current-voltage relationship of a solar cell is:[14]

$$I_L = I_S - I_O [\exp(eV/kT) - 1], \tag{11}$$

we may find V_{BASE} by setting I_L to zero and solving for V, which is then equal to V_{BASE}. This gives equation (12):

$$V_{BASE} = kT/e \ln(1+I_S/I_O) \quad (12)$$

$$= 0.774 \text{ V}$$

For example, a load voltage of 0.5 p.u. means that the output voltage of the cell is one-half of V_{BASE} or 0.387 V.

The normalized power is simply the product of the normalized voltage and the normalized current. The above value of V_{BASE} is close to the open-circuit cell voltages realized by researchers using high efficiency passivated emitter solar cells (PESC)[15].

For simulation purposes, resistance values are defined in terms of the open-circuit voltage of each individual circuit (V_{OC}) and the base current as given below.

$$R_{PU} = R/R_{BASE}, \quad (13)$$

where

$$R_{BASE} = V_{OC}/I_{BASE}. \quad (14)$$

V_{OC} is the open circuit voltage of the cell being simulated (in volts) at an insolation level of 1.0 p.u. It is important to note here that V_{OC} is only equal to V_{BASE} when the reference cell is being simulated. R_{PU} is given by the classical definition of per-unit resistance, and is used so that resistance values can be given relative to each circuit. A resistance value of 0.9 p.u. corresponds to a resistance of 69.6 Ω in a circuit using a cell with a I_O of 1.0E-13 p.u. and to a resistance of 37.5 Ω when using a cell with I_O equal to 1.0E-7 p.u.

Since both tracking and non-tracking arrays will be modeled, and the total amount of energy available differs in each case, two different base values will be used for energy. For the tracking arrays, the insolation is assumed to be constant for the length of the day, so the base value for energy is the maximum insolation in kW times the length of the day in seconds. For a 10 hr. day,

$$E_{BASE,T} = (1kW)\ (36000\ s) \quad (15)$$

$$= 36{,}000\ kJ$$

For non-tracking arrays, $E_{BASE,F}$ is given by integrating the insolation over the length of the day:

$$E_{BASE,F} = \int_0^T 1\ \sin(\pi t/T)\ dt\ \ kJ \quad (16)$$

$$= (2T/\pi)\ kJ$$

$$= 22{,}920\ kJ$$

where T is the length of the day in seconds.

Various cell fill-factors can be modeled by changing the dark current of the cell. The effect of series resistance is neglected, because for the purpose of the simulation, this resistance can be treated as a part of the resistive load. To take any series resistance into account, it is only necessary to use voltage-division.

Software Implementation

The 'MaxPU' procedure in the simulation program is used to calculate the cell output. The file(s) containing the normalized insolation data to be analyzed can be specified in the same manner as used by the 'DayGen' procedure. The normalized insolation values are then used as normalized values for the short-circuit current. Equation (11) is solved for I_L after replacing the voltage term by the product of the current and the load resistance, which results in equation (17). The output voltage, power, and energy values are then calculated using the load resistance. The 'R_{PU}' term in equation (17) represents either $R_{PU,MP}$ or $R_{PU,\eta}$, depending on whether the array is optimized to maximize power output or utilization efficiency.

$$I_L = I_S - I_O\,[\exp(eR_{PU}V_{OC}I_L/kT) - 1] \qquad (17)$$

The V_{OC} is present to convert the per-unit resistance to ohms. In order to solve equation (17) for I_L, it is necessary to use Newton's Method[16] in order to ensure convergence of the simulation program.

CHAPTER V

ANALYSIS

Simulation Plan

For both tracking and non-tracking arrays, fill factors ranging from about 0.55 to 0.85 were used in the simulations. This is done by varying I_O from 1.0E-2 I_S to 1.0E-13 I_S with each successive value being 1 or 2 orders of magnitude greater (or less) than the previous. Load resistances are set to maximize the instantaneous power output for both the tracking and fixed arrays ($R_{PU,MP}$). The load resistances were also set to maximize η_E when simulating the non-tracking arrays ($R_{PU,\eta}$). The derivation of the equation used to find $R_{PU,MP}$ is given in APPENDIX C. The equation governing $R_{PU,\eta}$ is very difficult to derive in closed form, and so the $R_{PU,\eta}$ values are determined by trial and error. The optimum resistance values and fill factors for each value of dark current are shown in Table 4.

TABLE 4

FILL FACTORS AND CORRESPONDING LOAD RESISTANCES

I_o	f.f.	$R_{PU,MP}$	$R_{PU,\eta}$
1.0E-13	0.8569	0.9225	1.092
1.0E-11	0.8381	0.9154	1.088
1.0E-9	0.8128	0.9067	1.084
1.0E-7	0.7766	0.8964	1.078
1.0E-5	0.7200	0.8852	1.070
1.0E-3	0.6179	0.8822	1.062
1.0E-2	0.5314	0.8995	1.062

Performance Criteria

Both the energy utilization efficiency and the total energy supplied during the day are used as performance criteria in this study. The utilization efficiency is most useful when comparing the effects of changing conditions on individual arrays. The values for energy supplied and percentage changes in η_E can be used to compare results between arrays. Using η_E by itself to compare results for different arrays tends to be misleading because the utilization efficiency is higher for low fill-factor arrays.

Simulation Program

The listing for the program used in this study is given in APPENDIX C, with a brief flow diagram shown in Figure 8. The program was written in Borland International's 'Turbo Pascal' and executed on an IBM PC/AT compatible computer. The program can be run from sets of menus or from text files containing the responses to the menu prompts.

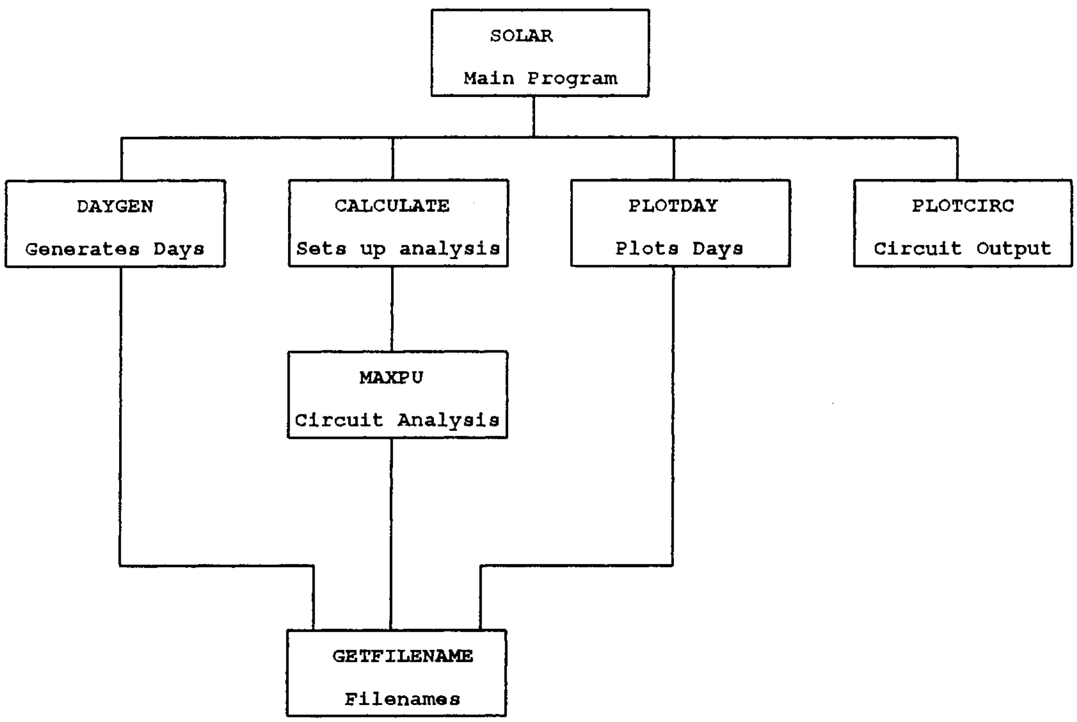

Figure 8. ˥low Diagram for Simulation Program

CHAPTER VI

DISCUSSION OF SIMULATION RESULTS

Tabulated results from the simulations are given in APPENDIX A, with plots of the results given in APPENDIX B.

Tracking Arrays vs. Fixed Arrays

The sensitivity of tracking arrays to the presence of clouds was much higher than that of the fixed arrays. While the efficiency was initially much higher with tracking arrays, as the cloudiness increased, the utilization efficiency dropped much faster than it did with the fixed arrays. After $\%_{CLOUD}$ increased to greater than 50%, both the total energy output and utilization efficiency of the fixed arrays were greater than with the tracking arrays. The relative efficiencies and energy outputs of the two systems when the fixed array is optimized for η_E are shown in tables 5, 6 and 7. The percent differences that are shown here are relative to the tracking array results.

TABLE 5

COMPARISON OF TRACKING ARRAYS VS. FIXED ARRAYS FOR CLEAR SKIES

I_O	E_N Tracking	E_N Fixed	%Diff.	η_E Tracking	η_E Fixed	%Diff.
1.0E-13	0.8569	0.7255	15.33	1.0000	0.8391	16.09
1.0E-11	0.7092	0.6038	14.86	1.0000	0.8426	15.74
1.0E-9	0.5627	0.4824	14.27	1.0000	0.8472	15.28
1.0E-7	0.4182	0.3616	13.53	1.0000	0.8539	14.61
1.0E-5	0.2769	0.2418	12.68	1.0000	0.8646	13.54
1.0E-3	0.1426	0.1252	12.20	1.0000	0.8872	11.28
1.0E-2	0.0818	0.0710	13.20	1.0000	0.9124	8.76

TABLE 6

COMPARISON OF TRACKING ARRAYS VS. FIXED ARRAYS FOR 30% COVER, 30% TRANSMISSIVITY

I_O	E_N Tracking	E_N Fixed	%Diff.	η_E Tracking	η_E Fixed	%Diff.
1.0E-13	0.6248	0.5670	9.25	0.9127	0.7806	14.47
1.0E-11	0.5173	0.4721	8.74	0.9125	0.7834	14.15
1.0E-9	0.4108	0.3775	8.11	0.9125	0.7873	13.72
1.0E-7	0.3057	0.2833	7.33	0.9132	0.7933	13.13
1.0E-5	0.2030	0.1899	6.45	0.9163	0.8045	12.20
1.0E-3	0.1051	0.0987	6.09	0.9287	0.8337	10.23
1.0E-2	0.0604	0.0561	7.12	0.9447	0.8715	7.75

TABLE 7

COMPARISON OF TRACKING ARRAYS VS. FIXED ARRAYS FOR 50% COVER, 30% TRANSMISSIVITY

I_O	E_N Tracking	E_N Fixed	%Diff.	η_E Tracking	η_E Fixed	%Diff.
1.0E-13	0.4700	0.5054	-7.53	0.8252	0.7522	8.85
1.0E-11	0.3894	0.4209	-8.09	0.8249	0.7546	8.52
1.0E-9	0.3096	0.3367	-8.75	0.8250	0.7581	8.11
1.0E-7	0.2308	0.2528	-9.53	0.8265	0.7638	7.59
1.0E-5	0.1538	0.1696	-10.27	0.8325	0.7750	6.91
1.0E-3	0.0802	0.0883	-10.10	0.8565	0.8070	5.78
1.0E-2	0.0462	0.0502	-8.66	0.8888	0.8507	4.29

The results shown in Table 7 are interesting because the fixed arrays actually supply more energy than the tracking arrays.

Optimization of Utilization Efficiency

Optimization of the utilization efficiency as opposed to the maximization of power output leads to utilization efficiency gains of about 1.6% for the low fill-factor array to about 4.5% for the highest fill factor. As the fill-factors decrease, there is less of an improvement in utilization efficiency. The gain in utilization efficiency and the decrease in the peak output power when the array is optimized for η_E as opposed to maximization of the peak output power are shown in Table 8. The percent differences are calculated relative to η_E.

TABLE 8

CHANGES IN η_E AND PEAK OUTPUT POWER

I_o	Change in Power			Change in η_E		
	η_E	Power	%Diff.	η_E	Power	%Diff.
1.0E-13	0.8009	0.8569	-6.99	0.8391	0.8016	4.47
1.0E-11	0.6655	0.7092	-6.57	0.8425	0.8055	4.39
1.0E-9	0.5304	0.5627	-6.09	0.8472	0.8112	4.25
1.0E-7	0.3968	0.4182	-5.39	0.8539	0.8198	3.99
1.0E-5	0.2652	0.2769	-4.41	0.8646	0.8345	3.48
1.0E-3	0.1386	0.1426	-2.89	0.8872	0.8653	2.47
1.0E-2	0.0803	0.0818	-1.87	0.9123	0.8979	1.58

Insolation Sensitivity by Fill Factor

As expected, the arrays with low fill factors are much less sensitive to clouds than arrays with higher fill factors. This is a result of the fact that I-V transfer curves are more rounded for cells with lower fill factors. This causes the locus of maximum power points to be closer to the load line. Plots of the output energy vs. $\%_{CLOUD}$ are shown in Figures 9. and 10. for fixed arrays. As the cloudiness increases, the low fill factor cells are less affected by the increase in cloudiness. In addition, it can also be seen here that energy output is less sensitive when η_E is maximized.

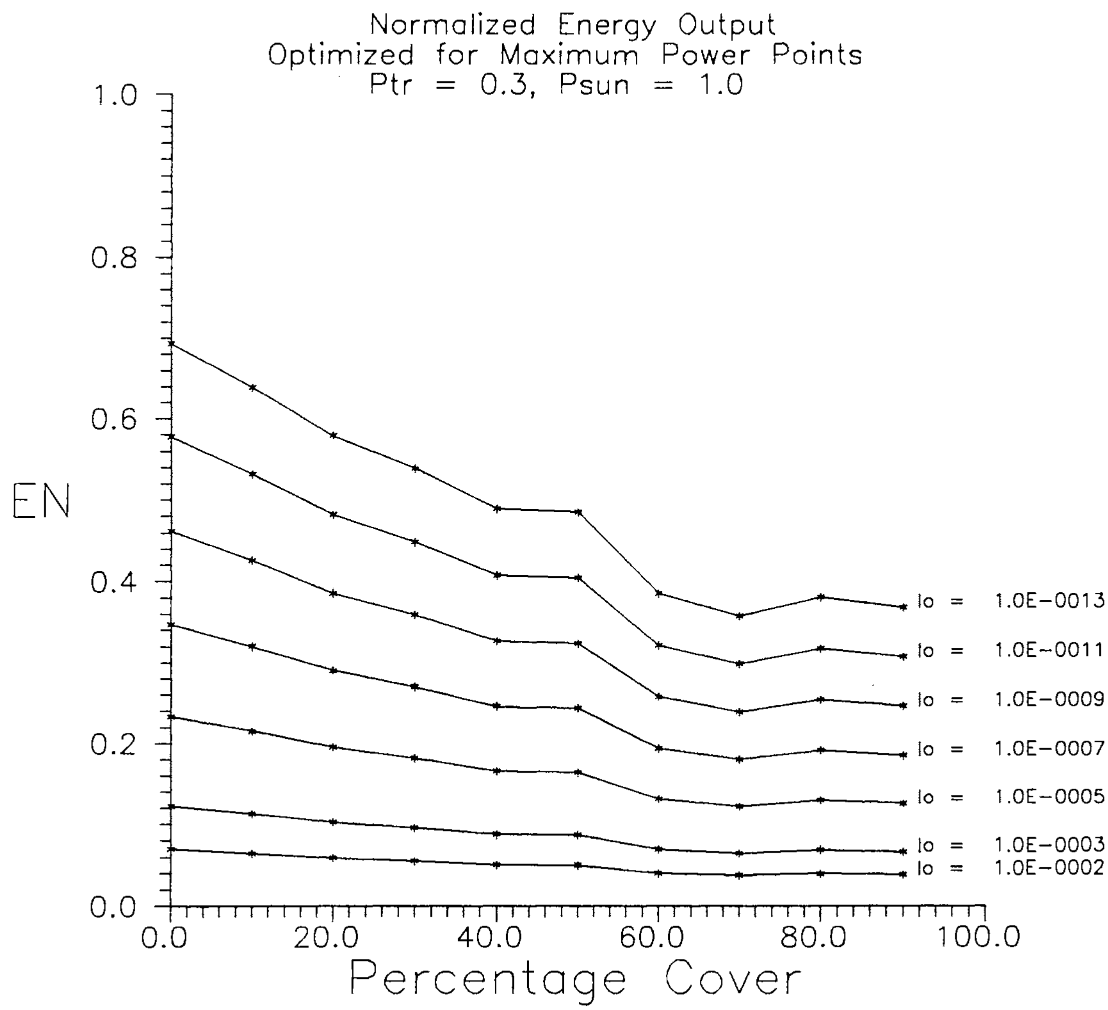

Figure 9. Fixed Array, Maximum Power Points

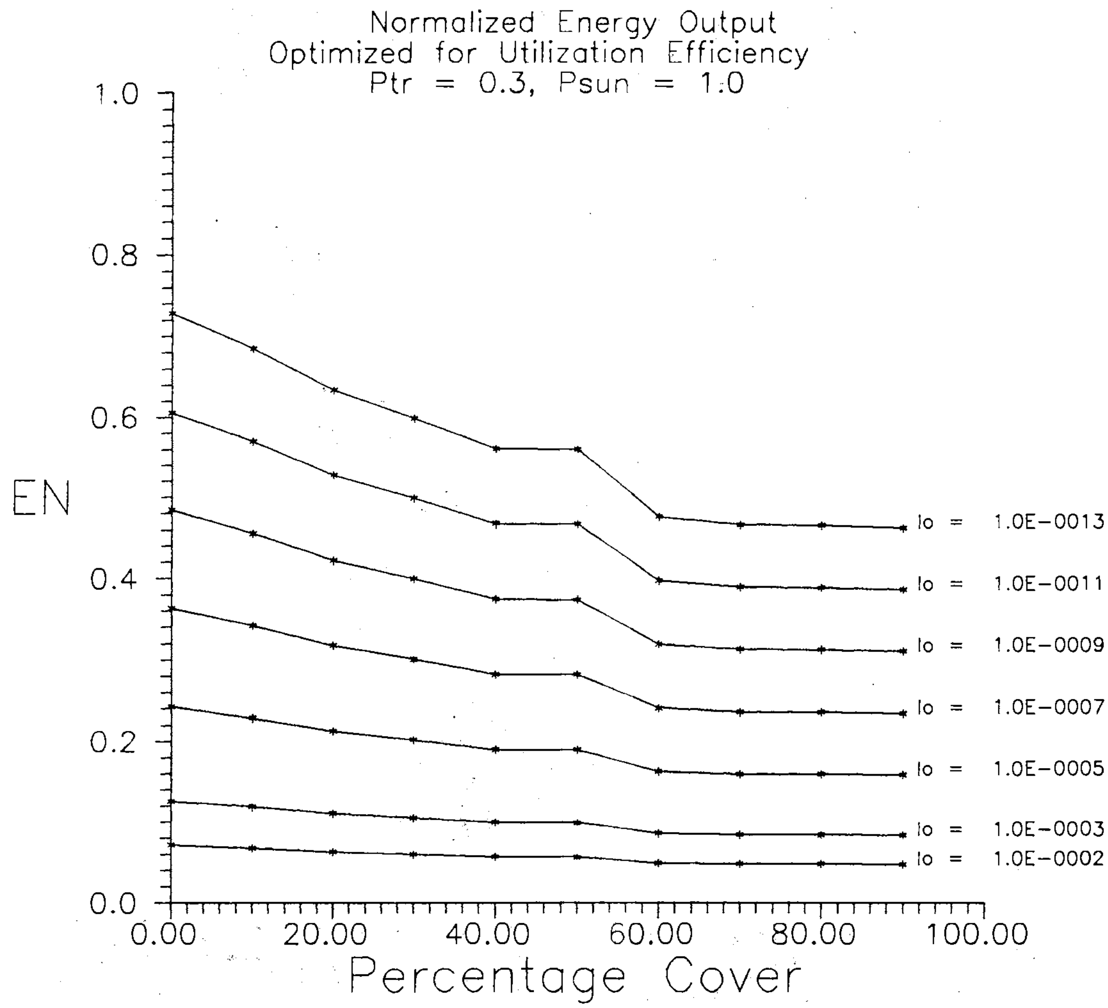

Figure 10. Fixed Array, Optimum Utilization Eff.

CHAPTER VII

CONCLUSIONS

General Results

In this paper, PV arrays were studied for a range of insolation conditions. The most startling results were the comparisons between the fixed arrays optimized for utilization efficiency and the tracking arrays when cloudy days were simulated. After a certain level of cloudiness was reached, the fixed arrays actually supplied more energy. Since the equipment necessary to add tracking capabilities to a PV array adds to both the initial and maintenance costs, in locations with more cloudy days, the additional expense of tracking hardware may not be worthwhile.

In addition, optimization of the arrays for utilization efficiency appears to be worthwhile in some cases. Less peak power is supplied by the array, but more energy is supplied of a day's time.

Suggestions for Further Study

1. In this study, the output power available from low fill-factor arrays was limited by the low open-circuit voltages. It is possible to simulate cells with low fill factors but with higher values of V_{oc} by including an 'ideality factor'[4] in equation (11). This factor is the 'A' term shown in

equation (18).

$$I_L = I_S - I_O\,[\exp(eV/AkT) - 1], \tag{18}$$

2. Simulate several real-world cells in addition to the ideal cells studied here.

3. Improve the insolation model to differentiate between one-axis tracking vs. two-axis tracking.

4. General improvement in the insolation models.

5. Simulate concentrating arrays.

6. Model different types of loads, for instance the water-electrolyzer load studied by Appelbaum.

BIBLIOGRAPHY

1. Glenn Zorpette, "Photovoltaics: technical gains and an uncertain market", *IEEE Spectrum*, Vol. 26, No. 7, pp. 42-43, July 1989

2. R. G. Ramakumar and Ward Jewell, *Public Service Company of Oklahoma Guide to Utility-Interactive Photovoltaic Systems*, October, 1984 p. 35

3. Michael W. Edenburn and Eldon C. Boes, "Photovoltaic Concentrators: Performance and Reliability Data and Future Design Directions", *Proceedings, 17th IEEE Photovoltaics Specialists Conference*, 1984, pp. 473-476

4. J. Appelbaum, "The Operation of Loads Powered by Single Sources or by a Common Source of Solar Cells", Copy is included with this paper, see APPENDIX D.

5. R. G. Ross Jr., et al, "Characterization of the Electrical Output of Flat-Plate Photovoltaic Arrays", *Proceedings, 16th IEEE Photovoltaics Specialists Conference*, 1982, pp. 384-390

6. R. G. Ross Jr., et al, "Photovoltaic Array - Power Conditioner Interface Characteristics", JPL Document No. 5101-202, Jet Propulsion Laboratory, Pasadena, California, December, 1982

7. M. A. Khallat and Saifur Rahman, "A Probabilistic Approach to Photovoltaic Generator Performance Prediction", *IEEE Transactions on Energy Conversion*, Vol. EC-1, No.3, pp. 34-40, September, 1986

8. M. A. Atwater and J. T. Ball, "A Numerical Solar Radiation Model Based On Standard Meteorological Observations", *Solar Energy*, Vol. 21, No 3, pp.163-170

9. J. E. Sherry and C. G. Justus, "A Simple Hourly All-Sky Solar Radiation Model Based on Meteorological Parameters", *Solar Energy*, Vol. 32, No. 2, pp. 195-204, 1984

10. B. H. Chowdhury, and S. Rahman, "Forecasting Sub-Hourly Irradiance for Prediction of Photovoltaic Output", *Proceedings, 19th IEEE Photovoltaics Specialists Conference*, pp. 171-174, 1987

11. M. K. El-Adawi, et al., "Estimation of the Hourly Solar Irradiance on a Horizontal Surface", *Solar Energy*, Vol. 36, No. 2, 1986, pp. 129-134

12. Ward T. Jewell, *The Effects of Moving Cloud Shadows on Electric Utilities with Dispersed Solar Photovoltaic Generation*, Ph.D. Dissertation, Oklahoma State University, 1988

13. R. G. Ross, et al, "Reference Conditions for Reporting Terrestrial Photovoltaic Performance", *Proceedings, 16th Annual IEEE Photovoltaics Specialists Conference*, pp. 384-390, 1982.

14. S. W. Angrist, *Direct Energy Conversion*, (4th ed.), p. 187, Boston: Allyn and Bacon, Inc., 1982

15. M. A. Green, S. R. Wenham, and A. W. Blakers, "Recent Advances in High Efficiency Silicon Solar Cells", *Proceedings, 19th IEEE Photovoltaic Specialists Conference*, 1987, pp. 6-12

16. Curtis F. Gerald and Patrick O. Wheatley, *Applied Numerical Analysis* (3rd ed.). pp. 15-20, Addison-Wesley Publishing Co., 1984

APPENDICES

APPENDIX A

TABULATED SIMULATION DATA

TABLE 9

TRACKING ARRAY
$\%_{TR} = 30$, $\%_{CLEAR} = 100$

Io = 1.0E-0013, R = 0.923, %Tr = 30, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.8569	0.8569	1.0000	0.0000
10	0.7796	0.7995	0.9751	0.0199
20	0.7022	0.7420	0.9463	0.0398
30	0.6248	0.6845	0.9127	0.0597
40	0.5474	0.6270	0.8730	0.0796
50	0.4700	0.5695	0.8252	0.0995
60	0.3926	0.5120	0.7667	0.1194
70	0.3152	0.4546	0.6934	0.1394
80	0.2378	0.3971	0.5989	0.1593
90	0.1604	0.3396	0.4724	0.1792

Io = 1.0E-0011, R = 0.915, %Tr = 30, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.7092	0.7092	1.0000	0.0000
10	0.6452	0.6618	0.9750	0.0165
20	0.5813	0.6144	0.9462	0.0331
30	0.5173	0.5669	0.9125	0.0496
40	0.4534	0.5195	0.8727	0.0661
50	0.3894	0.4721	0.8249	0.0827
60	0.3255	0.4247	0.7664	0.0992
70	0.2615	0.3773	0.6932	0.1158
80	0.1976	0.3299	0.5990	0.1323
90	0.1337	0.2825	0.4731	0.1488

TABLE 9 (Continued)

Io = 1.0E-0009, R = 0.907, %Tr = 30, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.5627	0.5627	1.0000	0.0000
10	0.5121	0.5252	0.9750	0.0131
20	0.4614	0.4877	0.9461	0.0263
30	0.4108	0.4502	0.9125	0.0394
40	0.3602	0.4128	0.8727	0.0525
50	0.3096	0.3753	0.8250	0.0657
60	0.2590	0.3378	0.7667	0.0788
70	0.2084	0.3003	0.6938	0.0920
80	0.1577	0.2628	0.6002	0.1051
90	0.1071	0.2253	0.4753	0.1182

Io = 1.0E-0007, R = 0.896, %Tr = 30, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.4182	0.4182	1.0000	0.0000
10	0.3807	0.3904	0.9752	0.0097
20	0.3432	0.3626	0.9465	0.0194
30	0.3057	0.3348	0.9132	0.0291
40	0.2683	0.3070	0.8737	0.0388
50	0.2308	0.2792	0.8265	0.0485
60	0.1933	0.2515	0.7688	0.0581
70	0.1559	0.2237	0.6967	0.0678
80	0.1184	0.1959	0.6043	0.0775
90	0.0809	0.1681	0.4812	0.0872

TABLE 9 (Continued)

Io = 1.0E-0005, R = 0.885, %Tr = 30, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.2769	0.2769	1.0000	0.0000
10	0.2523	0.2585	0.9761	0.0062
20	0.2277	0.2400	0.9485	0.0124
30	0.2030	0.2216	0.9163	0.0186
40	0.1784	0.2031	0.8782	0.0247
50	0.1538	0.1847	0.8325	0.0309
60	0.1291	0.1662	0.7767	0.0371
70	0.1045	0.1478	0.7070	0.0433
80	0.0799	0.1294	0.6174	0.0495
90	0.0552	0.1109	0.4980	0.0557

Io = 1.0E-0003, R = 0.882, %Tr = 30, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.1426	0.1426	0.9999	0.0000
10	0.1301	0.1328	0.9797	0.0027
20	0.1176	0.1230	0.9562	0.0054
30	0.1051	0.1132	0.9287	0.0081
40	0.0926	0.1034	0.8961	0.0107
50	0.0802	0.0936	0.8565	0.0134
60	0.0677	0.0838	0.8077	0.0161
70	0.0552	0.0740	0.7460	0.0188
80	0.0427	0.0642	0.6653	0.0215
90	0.0302	0.0544	0.5556	0.0242

TABLE 9 (Continued)

Io = 1.0E-0002, R = 0.900, %Tr = 30, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.0818	0.0819	0.9978	0.0002
10	0.0746	0.0759	0.9829	0.0013
20	0.0675	0.0700	0.9654	0.0024
30	0.0604	0.0640	0.9447	0.0035
40	0.0533	0.0580	0.9196	0.0047
50	0.0462	0.0520	0.8888	0.0058
60	0.0391	0.0460	0.8500	0.0069
70	0.0320	0.0401	0.7997	0.0080
80	0.0249	0.0341	0.7316	0.0091
90	0.0178	0.0281	0.6346	0.0103

TABLE 10

FIXED ARRAY, OPTIMIZED FOR POWER
$\%_{TR} = 30$, $\%_{CLEAR} = 100$

Io = 1.0E-0013, R = 0.923, %Tr = 30, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.6931	0.8647	0.8016	0.1716
10	0.6412	0.8179	0.7839	0.1767
20	0.5774	0.7598	0.7599	0.1824
30	0.5406	0.7263	0.7443	0.1857
40	0.4832	0.6745	0.7164	0.1913
50	0.4807	0.6719	0.7155	0.1911
60	0.3851	0.5854	0.6577	0.2004
70	0.3590	0.5619	0.6389	0.2029
80	0.3675	0.5708	0.6439	0.2033
90	0.3678	0.5676	0.6481	0.1997

Io = 1.0E-0011, R = 0.915, %Tr = 30, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.5773	0.7167	0.8056	0.1393
10	0.5341	0.6781	0.7876	0.1440
20	0.4810	0.6303	0.7632	0.1493
30	0.4503	0.6026	0.7473	0.1523
40	0.4026	0.5599	0.7190	0.1573
50	0.4005	0.5578	0.7181	0.1573
60	0.3209	0.4866	0.6596	0.1656
70	0.2992	0.4672	0.6405	0.1680
80	0.3063	0.4745	0.6457	0.1681
90	0.3066	0.4718	0.6498	0.1652

TABLE 10 (Continued)

Io = 1.0E-0009, R = 0.907, %Tr = 30, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.4619	0.5694	0.8112	0.1075
10	0.4273	0.5390	0.7928	0.1117
20	0.3849	0.5013	0.7679	0.1164
30	0.3604	0.4795	0.7516	0.1191
40	0.3223	0.4458	0.7230	0.1235
50	0.3206	0.4441	0.7220	0.1235
60	0.2571	0.3879	0.6627	0.1308
70	0.2397	0.3726	0.6433	0.1329
80	0.2454	0.3783	0.6487	0.1329
90	0.2456	0.3763	0.6527	0.1307

Io = 1.0E-0007, R = 0.896, %Tr = 30, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.3472	0.4235	0.8198	0.0763
10	0.3212	0.4010	0.8010	0.0798
20	0.2894	0.3732	0.7756	0.0838
30	0.2710	0.3571	0.7590	0.0861
40	0.2425	0.3322	0.7300	0.0897
50	0.2412	0.3310	0.7288	0.0898
60	0.1936	0.2895	0.6687	0.0959
70	0.1805	0.2782	0.6489	0.0977
80	0.1849	0.2824	0.6547	0.0975
90	0.1850	0.2809	0.6585	0.0959

TABLE 10 (Continued)

Io = 1.0E-0005, R = 0.885, %Tr = 30, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.2334	0.2797	0.8345	0.0463
10	0.2161	0.2649	0.8154	0.0489
20	0.1948	0.2466	0.7898	0.0518
30	0.1824	0.2360	0.7730	0.0536
40	0.1634	0.2196	0.7439	0.0562
50	0.1625	0.2189	0.7426	0.0563
60	0.1307	0.1915	0.6823	0.0609
70	0.1219	0.1841	0.6622	0.0622
80	0.1249	0.1869	0.6684	0.0620
90	0.1249	0.1859	0.6717	0.0610

Io = 1.0E-0003, R = 0.882, %Tr = 30, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.1221	0.1411	0.8653	0.0190
10	0.1131	0.1334	0.8479	0.0203
20	0.1021	0.1239	0.8244	0.0218
30	0.0958	0.1184	0.8089	0.0226
40	0.0859	0.1099	0.7822	0.0239
50	0.0855	0.1095	0.7807	0.0240
60	0.0690	0.0953	0.7248	0.0262
70	0.0645	0.0914	0.7058	0.0269
80	0.0661	0.0928	0.7121	0.0267
90	0.0660	0.0924	0.7147	0.0263

TABLE 10 (Continued)

Io = 1.0E-0002, R = 0.900, %Tr = 30, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.0699	0.0778	0.8979	0.0079
10	0.0648	0.0733	0.8844	0.0085
20	0.0586	0.0676	0.8663	0.0090
30	0.0550	0.0644	0.8541	0.0094
40	0.0494	0.0593	0.8330	0.0099
50	0.0491	0.0591	0.8317	0.0099
60	0.0398	0.0506	0.7863	0.0108
70	0.0372	0.0483	0.7705	0.0111
80	0.0381	0.0491	0.7758	0.0110
90	0.0381	0.0489	0.7782	0.0108

TABLE 11

FIXED ARRAY, OPTIMIZED FOR η_E
$\%_{TR}$ = 30, $\%_{CLEAR}$ = 100

Io = 1.0E-0013, R = 1.092, %Tr = 30, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.7255	0.8647	0.8391	0.1391
10	0.6714	0.8179	0.8208	0.1465
20	0.6054	0.7598	0.7967	0.1545
30	0.5670	0.7263	0.7806	0.1593
40	0.5086	0.6745	0.7541	0.1658
50	0.5054	0.6719	0.7522	0.1665
60	0.4073	0.5854	0.6957	0.1781
70	0.3798	0.5619	0.6760	0.1821
80	0.3896	0.5708	0.6826	0.1812
90	0.3895	0.5676	0.6863	0.1780

Io = 1.0E-0011, R = 1.088, %Tr = 30, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.6038	0.7167	0.8426	0.1128
10	0.5588	0.6781	0.8241	0.1193
20	0.5040	0.6303	0.7996	0.1263
30	0.4721	0.6026	0.7834	0.1305
40	0.4236	0.5599	0.7565	0.1363
50	0.4209	0.5578	0.7546	0.1369
60	0.3394	0.4866	0.6976	0.1472
70	0.3165	0.4672	0.6775	0.1507
80	0.3247	0.4745	0.6844	0.1498
90	0.3245	0.4718	0.6878	0.1473

TABLE 11 (Continued)

Io = 1.0E-0009, R = 1.084, %Tr = 30, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.4825	0.5694	0.8472	0.0870
10	0.4466	0.5390	0.8285	0.0925
20	0.4029	0.5013	0.8037	0.0984
30	0.3775	0.4795	0.7873	0.1020
40	0.3388	0.4458	0.7600	0.1070
50	0.3367	0.4441	0.7581	0.1074
60	0.2717	0.3879	0.7005	0.1162
70	0.2534	0.3726	0.6802	0.1192
80	0.2600	0.3783	0.6873	0.1183
90	0.2598	0.3763	0.6905	0.1165

Io = 1.0E-0007, R = 1.078, %Tr = 30, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.3616	0.4235	0.8539	0.0619
10	0.3348	0.4010	0.8349	0.0662
20	0.3022	0.3732	0.8099	0.0710
30	0.2833	0.3571	0.7933	0.0738
40	0.2543	0.3322	0.7657	0.0778
50	0.2528	0.3310	0.7638	0.0782
60	0.2043	0.2895	0.7057	0.0852
70	0.1906	0.2782	0.6852	0.0876
80	0.1956	0.2824	0.6926	0.0868
90	0.1953	0.2809	0.6953	0.0856

TABLE 11 (Continued)

Io = 1.0E-0005, R = 1.070, %Tr = 30, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.2418	0.2797	0.8646	0.0379
10	0.2241	0.2649	0.8459	0.0408
20	0.2025	0.2466	0.8210	0.0442
30	0.1899	0.2360	0.8045	0.0461
40	0.1707	0.2196	0.7770	0.0490
50	0.1696	0.2189	0.7750	0.0492
60	0.1374	0.1915	0.7174	0.0541
70	0.1284	0.1841	0.6972	0.0558
80	0.1317	0.1869	0.7047	0.0552
90	0.1314	0.1859	0.7069	0.0545

Io = 1.0E-0003, R = 1.062, %Tr = 30, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.1252	0.1411	0.8872	0.0159
10	0.1161	0.1334	0.8706	0.0173
20	0.1051	0.1239	0.8484	0.0188
30	0.0987	0.1184	0.8337	0.0197
40	0.0889	0.1099	0.8089	0.0210
50	0.0883	0.1095	0.8070	0.0211
60	0.0719	0.0953	0.7548	0.0234
70	0.0673	0.0914	0.7365	0.0241
80	0.0690	0.0928	0.7434	0.0238
90	0.0688	0.0924	0.7450	0.0235

TABLE 11 (Continued)

Io = 1.0E-0002, R = 1.062, %Tr = 30, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.0710	0.0778	0.9124	0.0068
10	0.0659	0.0733	0.8998	0.0073
20	0.0597	0.0676	0.8829	0.0079
30	0.0561	0.0644	0.8715	0.0083
40	0.0505	0.0593	0.8522	0.0088
50	0.0502	0.0591	0.8507	0.0088
60	0.0409	0.0506	0.8089	0.0097
70	0.0384	0.0483	0.7940	0.0100
80	0.0393	0.0491	0.7994	0.0099
90	0.0392	0.0489	0.8013	0.0097

TABLE 12

TRACKING ARRAY
$\%_{TR}$ = 50, $\%_{CLEAR}$ = 100

Io = 1.0E-0013, R = 0.923, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.8569	0.8569	1.0000	0.0000
10	0.7943	0.8160	0.9735	0.0216
20	0.7317	0.7750	0.9441	0.0433
30	0.6690	0.7340	0.9115	0.0649
40	0.6064	0.6930	0.8751	0.0866
50	0.5438	0.6520	0.8340	0.1082
60	0.4812	0.6110	0.7874	0.1299
70	0.4185	0.5700	0.7342	0.1515
80	0.3559	0.5291	0.6727	0.1732
90	0.2933	0.4881	0.6009	0.1948

Io = 1.0E-0011, R = 0.915, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.7092	0.7092	1.0000	0.0000
10	0.6576	0.6753	0.9738	0.0177
20	0.6061	0.6415	0.9448	0.0354
30	0.5545	0.6077	0.9125	0.0531
40	0.5030	0.5738	0.8765	0.0709
50	0.4514	0.5400	0.8360	0.0886
60	0.3999	0.5062	0.7900	0.1063
70	0.3483	0.4723	0.7374	0.1240
80	0.2967	0.4385	0.6768	0.1417
90	0.2452	0.4046	0.6060	0.1594

TABLE 12 (Continued)

Io = 1.0E-0009, R = 0.907, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.5627	0.5627	1.0000	0.0000
10	0.5221	0.5359	0.9743	0.0138
20	0.4815	0.5091	0.9459	0.0276
30	0.4410	0.4823	0.9143	0.0413
40	0.4004	0.4555	0.8790	0.0551
50	0.3598	0.4287	0.8393	0.0689
60	0.3192	0.4019	0.7943	0.0827
70	0.2786	0.3751	0.7429	0.0964
80	0.2381	0.3483	0.6835	0.1102
90	0.1975	0.3215	0.6143	0.1240

Io = 1.0E-0007, R = 0.896, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.4182	0.4182	1.0000	-0.0000
10	0.3884	0.3982	0.9753	0.0098
20	0.3586	0.3783	0.9480	0.0197
30	0.3289	0.3584	0.9177	0.0295
40	0.2991	0.3384	0.8838	0.0393
50	0.2694	0.3185	0.8457	0.0491
60	0.2396	0.2986	0.8025	0.0590
70	0.2099	0.2787	0.7531	0.0688
80	0.1801	0.2587	0.6961	0.0786
90	0.1504	0.2388	0.6296	0.0885

TABLE 12 (Continued)

Io = 1.0E-0005, R = 0.885, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.2769	0.2769	1.0000	0.0000
10	0.2577	0.2636	0.9775	0.0059
20	0.2385	0.2503	0.9527	0.0118
30	0.2192	0.2370	0.9251	0.0178
40	0.2000	0.2237	0.8942	0.0237
50	0.1808	0.2103	0.8593	0.0296
60	0.1615	0.1970	0.8198	0.0355
70	0.1423	0.1837	0.7745	0.0414
80	0.1230	0.1704	0.7221	0.0473
90	0.1038	0.1571	0.6609	0.0533

Io = 1.0E-0003, R = 0.882, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.1426	0.1426	0.9999	0.0000
10	0.1331	0.1354	0.9826	0.0024
20	0.1235	0.1282	0.9634	0.0047
30	0.1140	0.1211	0.9419	0.0070
40	0.1045	0.1139	0.9177	0.0094
50	0.0950	0.1067	0.8903	0.0117
60	0.0854	0.0995	0.8589	0.0140
70	0.0759	0.0923	0.8225	0.0164
80	0.0664	0.0851	0.7801	0.0187
90	0.0569	0.0779	0.7298	0.0211

TABLE 12 (Continued)

Io = 1.0E-0002, R = 0.900, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.0818	0.0819	0.9978	0.0002
10	0.0763	0.0775	0.9854	0.0011
20	0.0709	0.0730	0.9715	0.0021
30	0.0655	0.0685	0.9557	0.0030
40	0.0601	0.0641	0.9377	0.0040
50	0.0547	0.0596	0.9171	0.0049
60	0.0493	0.0552	0.8931	0.0059
70	0.0439	0.0507	0.8649	0.0069
80	0.0384	0.0462	0.8312	0.0078
90	0.0330	0.0418	0.7904	0.0088

TABLE 13

FIXED ARRAY, OPTIMIZED FOR POWER
$\%_{TR}$ = 50, $\%_{CLEAR}$ = 100

Io = 1.0E-0013, R = 0.923, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.6953	0.8678	0.8012	0.1725
10	0.6554	0.8366	0.7834	0.1812
20	0.6006	0.7928	0.7576	0.1922
30	0.5709	0.7701	0.7413	0.1993
40	0.5249	0.7339	0.7153	0.2090
50	0.5304	0.7377	0.7190	0.2073
60	0.4438	0.6697	0.6627	0.2259
70	0.4313	0.6598	0.6537	0.2285
80	0.4246	0.6547	0.6484	0.2302
90	0.4345	0.6624	0.6560	0.2279

Io = 1.0E-0011, R = 0.915, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.5792	0.7192	0.8053	0.1400
10	0.5460	0.6935	0.7873	0.1475
20	0.5005	0.6574	0.7612	0.1570
30	0.4758	0.6387	0.7449	0.1629
40	0.4377	0.6088	0.7189	0.1712
50	0.4422	0.6120	0.7226	0.1698
60	0.3702	0.5559	0.6660	0.1857
70	0.3598	0.5477	0.6570	0.1879
80	0.3543	0.5436	0.6517	0.1893
90	0.3625	0.5498	0.6593	0.1873

TABLE 13 (Continued)

Io = 1.0E-0009, R = 0.907, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.4634	0.5715	0.8110	0.1080
10	0.4369	0.5511	0.7928	0.1142
20	0.4007	0.5226	0.7666	0.1220
30	0.3810	0.5078	0.7503	0.1268
40	0.3507	0.4842	0.7243	0.1335
50	0.3543	0.4867	0.7279	0.1324
60	0.2969	0.4424	0.6712	0.1455
70	0.2887	0.4360	0.6621	0.1473
80	0.2843	0.4327	0.6569	0.1484
90	0.2909	0.4377	0.6646	0.1468

Io = 1.0E-0007, R = 0.896, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.3483	0.4250	0.8197	0.0766
10	0.3285	0.4099	0.8014	0.0814
20	0.3015	0.3889	0.7753	0.0874
30	0.2869	0.3779	0.7592	0.0910
40	0.2644	0.3604	0.7334	0.0961
50	0.2670	0.3623	0.7369	0.0953
60	0.2242	0.3295	0.6804	0.1053
70	0.2180	0.3247	0.6713	0.1067
80	0.2147	0.3223	0.6663	0.1076
90	0.2197	0.3260	0.6741	0.1062

TABLE 13 (Continued)

Io = 1.0E-0005, R = 0.885, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.2342	0.2807	0.8346	0.0464
10	0.2210	0.2707	0.8165	0.0497
20	0.2032	0.2568	0.7914	0.0536
30	0.1936	0.2495	0.7758	0.0559
40	0.1787	0.2380	0.7510	0.0592
50	0.1804	0.2392	0.7542	0.0588
60	0.1521	0.2175	0.6994	0.0654
70	0.1480	0.2144	0.6904	0.0664
80	0.1459	0.2128	0.6857	0.0669
90	0.1492	0.2152	0.6934	0.0660

Io = 1.0E-0003, R = 0.882, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.1225	0.1416	0.8656	0.0190
10	0.1159	0.1363	0.8499	0.0205
20	0.1068	0.1290	0.8282	0.0222
30	0.1020	0.1252	0.8149	0.0232
40	0.0945	0.1191	0.7935	0.0246
50	0.0953	0.1197	0.7961	0.0244
60	0.0810	0.1083	0.7480	0.0273
70	0.0789	0.1067	0.7400	0.0277
80	0.0779	0.1058	0.7358	0.0280
90	0.0796	0.1071	0.7429	0.0275

TABLE 13 (Continued)

Io = 1.0E-0002, R = 0.900, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.0702	0.0781	0.8981	0.0080
10	0.0664	0.0750	0.8859	0.0086
20	0.0613	0.0705	0.8690	0.0092
30	0.0586	0.0682	0.8587	0.0096
40	0.0543	0.0646	0.8418	0.0102
50	0.0548	0.0650	0.8438	0.0101
60	0.0468	0.0581	0.8051	0.0113
70	0.0456	0.0571	0.7985	0.0115
80	0.0450	0.0566	0.7951	0.0116
90	0.0459	0.0573	0.8010	0.0114

TABLE 14

FIXED ARRAY, OPTIMIZED FOR η_E
$\%_{TR}$ = 50, $\%_{CLEAR}$ = 100

Io = 1.0E-0013, R = 1.092, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.7282	0.8678	0.8391	0.1396
10	0.6847	0.8342	0.8209	0.1494
20	0.6331	0.7929	0.7985	0.1597
30	0.5985	0.7662	0.7812	0.1677
40	0.5606	0.7360	0.7617	0.1754
50	0.5599	0.7354	0.7613	0.1755
60	0.4759	0.6695	0.7109	0.1935
70	0.4659	0.6617	0.7041	0.1958
80	0.4653	0.6611	0.7038	0.1958
90	0.4621	0.6587	0.7015	0.1966

Io = 1.0E-0011, R = 1.088, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.6061	0.7192	0.8427	0.1132
10	0.5700	0.6915	0.8242	0.1215
20	0.5274	0.6575	0.8022	0.1300
30	0.4988	0.6355	0.7849	0.1367
40	0.4674	0.6106	0.7654	0.1432
50	0.4667	0.6100	0.7651	0.1433
60	0.3973	0.5557	0.7149	0.1585
70	0.3889	0.5494	0.7080	0.1604
80	0.3885	0.5488	0.7079	0.1603
90	0.3858	0.5469	0.7055	0.1611

TABLE 14 (Continued)

Io = 1.0E-0009, R = 1.084, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.4843	0.5715	0.8474	0.0872
10	0.4555	0.5496	0.8289	0.0940
20	0.4220	0.5227	0.8074	0.1007
30	0.3993	0.5053	0.7902	0.1060
40	0.3744	0.4856	0.7710	0.1112
50	0.3739	0.4852	0.7705	0.1113
60	0.3189	0.4423	0.7209	0.1234
70	0.3123	0.4373	0.7141	0.1250
80	0.3120	0.4368	0.7142	0.1249
90	0.3098	0.4353	0.7117	0.1255

Io = 1.0E-0007, R = 1.078, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.3630	0.4250	0.8542	0.0620
10	0.3417	0.4088	0.8359	0.0671
20	0.3170	0.3889	0.8151	0.0719
30	0.3002	0.3760	0.7983	0.0758
40	0.2818	0.3615	0.7796	0.0797
50	0.2814	0.3612	0.7791	0.0798
60	0.2407	0.3294	0.7307	0.0887
70	0.2358	0.3257	0.7241	0.0899
80	0.2356	0.3253	0.7243	0.0897
90	0.2340	0.3242	0.7217	0.0902

TABLE 14 (Continued)

Io = 1.0E-0005, R = 1.070, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.2428	0.2807	0.8651	0.0378
10	0.2289	0.2700	0.8477	0.0411
20	0.2127	0.2568	0.8284	0.0441
30	0.2018	0.2483	0.8126	0.0465
40	0.1898	0.2387	0.7950	0.0489
50	0.1895	0.2385	0.7945	0.0490
60	0.1629	0.2175	0.7490	0.0546
70	0.1597	0.2150	0.7429	0.0553
80	0.1596	0.2148	0.7432	0.0551
90	0.1585	0.2140	0.7406	0.0555

Io = 1.0E-0003, R = 1.062, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.1257	0.1416	0.8878	0.0159
10	0.1187	0.1359	0.8731	0.0173
20	0.1106	0.1290	0.8571	0.0184
30	0.1051	0.1245	0.8438	0.0195
40	0.0990	0.1195	0.8289	0.0204
50	0.0989	0.1194	0.8283	0.0205
60	0.0855	0.1083	0.7896	0.0228
70	0.0840	0.1070	0.7846	0.0231
80	0.0839	0.1068	0.7848	0.0230
90	0.0833	0.1065	0.7824	0.0232

TABLE 14 (Continued)

Io = 1.0E-0002, R = 1.062, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.0713	0.0781	0.9128	0.0068
10	0.0674	0.0747	0.9014	0.0074
20	0.0627	0.0706	0.8889	0.0078
30	0.0596	0.0678	0.8783	0.0083
40	0.0561	0.0648	0.8664	0.0087
50	0.0561	0.0647	0.8660	0.0087
60	0.0484	0.0580	0.8346	0.0096
70	0.0476	0.0573	0.8305	0.0097
80	0.0475	0.0572	0.8306	0.0097
90	0.0472	0.0569	0.8285	0.0098

TABLE 15

TRACKING ARRAY
$\%_{TR}$ = 30, $\%_{CLEAR}$ = 80

Io = 1.0E-0013, R = 0.923, %Tr = 30, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.5898	0.6928	0.8514	0.1029
10	0.5362	0.6467	0.8291	0.1105
20	0.4825	0.6006	0.8034	0.1181
30	0.4288	0.5545	0.7734	0.1257
40	0.3752	0.5084	0.7379	0.1332
50	0.3215	0.4623	0.6954	0.1408
60	0.2678	0.4162	0.6434	0.1484
70	0.2141	0.3701	0.5786	0.1560
80	0.1605	0.3240	0.4952	0.1636
90	0.1068	0.2780	0.3843	0.1711

Io = 1.0E-0011, R = 0.915, %Tr = 30, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.4944	0.5734	0.8621	0.0790
10	0.4494	0.5354	0.8393	0.0861
20	0.4044	0.4975	0.8129	0.0931
30	0.3594	0.4595	0.7822	0.1001
40	0.3145	0.4215	0.7460	0.1071
50	0.2695	0.3836	0.7025	0.1141
60	0.2245	0.3456	0.6496	0.1211
70	0.1795	0.3077	0.5836	0.1281
80	0.1346	0.2697	0.4989	0.1351
90	0.0896	0.2317	0.3866	0.1421

TABLE 15 (Continued)

Io = 1.0E-0009, R = 0.907, %Tr = 30, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.3986	0.4549	0.8762	0.0563
10	0.3623	0.4249	0.8527	0.0626
20	0.3261	0.3950	0.8256	0.0689
30	0.2898	0.3650	0.7940	0.0752
40	0.2536	0.3351	0.7568	0.0815
50	0.2174	0.3051	0.7123	0.0878
60	0.1811	0.2752	0.6582	0.0941
70	0.1449	0.2452	0.5908	0.1004
80	0.1086	0.2153	0.5046	0.1067
90	0.0724	0.1853	0.3906	0.1129

Io = 1.0E-0007, R = 0.896, %Tr = 30, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.3019	0.3376	0.8941	0.0357
10	0.2745	0.3155	0.8699	0.0410
20	0.2471	0.2934	0.8420	0.0463
30	0.2197	0.2713	0.8096	0.0516
40	0.1922	0.2492	0.7715	0.0570
50	0.1648	0.2271	0.7259	0.0623
60	0.1374	0.2050	0.6704	0.0676
70	0.1100	0.1829	0.6016	0.0729
80	0.0826	0.1608	0.5138	0.0782
90	0.0552	0.1387	0.3981	0.0835

TABLE 15 (Continued)

Io = 1.0E-0005, R = 0.885, %Tr = 30, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.2037	0.2226	0.9150	0.0189
10	0.1853	0.2081	0.8905	0.0228
20	0.1669	0.1935	0.8624	0.0266
30	0.1485	0.1790	0.8296	0.0305
40	0.1301	0.1644	0.7911	0.0343
50	0.1116	0.1498	0.7450	0.0382
60	0.0932	0.1353	0.6891	0.0421
70	0.0748	0.1207	0.6197	0.0459
80	0.0564	0.1062	0.5313	0.0498
90	0.0380	0.0916	0.4147	0.0536

Io = 1.0E-0003, R = 0.882, %Tr = 30, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.1057	0.1127	0.9377	0.0070
10	0.0962	0.1051	0.9156	0.0089
20	0.0868	0.0975	0.8901	0.0107
30	0.0774	0.0899	0.8603	0.0126
40	0.0680	0.0824	0.8251	0.0144
50	0.0585	0.0748	0.7827	0.0163
60	0.0491	0.0672	0.7307	0.0181
70	0.0397	0.0596	0.6655	0.0199
80	0.0303	0.0521	0.5813	0.0218
90	0.0208	0.0445	0.4685	0.0236

TABLE 15 (Continued)

Io = 1.0E-0002, R = 0.900, %Tr = 30, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.0599	0.0629	0.9517	0.0030
10	0.0546	0.0584	0.9347	0.0038
20	0.0493	0.0539	0.9148	0.0046
30	0.0440	0.0494	0.8913	0.0054
40	0.0387	0.0449	0.8630	0.0061
50	0.0334	0.0404	0.8285	0.0069
60	0.0281	0.0358	0.7852	0.0077
70	0.0229	0.0313	0.7295	0.0085
80	0.0176	0.0268	0.6550	0.0093
90	0.0123	0.0223	0.5504	0.0100

TABLE 16

FIXED ARRAY, OPTIMIZED FOR POWER
$\%_{TR} = 30$, $\%_{CLEAR} = 80$

Io = 1.0E-0013, R = 0.923, %Tr = 30, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.4555	0.7023	0.6487	0.2467
10	0.4214	0.6646	0.6341	0.2432
20	0.3800	0.6182	0.6147	0.2382
30	0.3574	0.5918	0.6039	0.2344
40	0.3178	0.5483	0.5796	0.2305
50	0.3179	0.5503	0.5777	0.2324
60	0.2524	0.4773	0.5287	0.2250
70	0.2390	0.4616	0.5178	0.2226
80	0.2522	0.4758	0.5300	0.2236
90	0.2444	0.4686	0.5216	0.2242

Io = 1.0E-0011, R = 0.915, %Tr = 30, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.3822	0.5826	0.6561	0.2004
10	0.3536	0.5516	0.6410	0.1980
20	0.3188	0.5134	0.6211	0.1945
30	0.2999	0.4917	0.6100	0.1918
40	0.2666	0.4558	0.5850	0.1892
50	0.2668	0.4575	0.5831	0.1907
60	0.2118	0.3974	0.5329	0.1856
70	0.2006	0.3844	0.5218	0.1839
80	0.2116	0.3961	0.5342	0.1845
90	0.2051	0.3902	0.5256	0.1851

TABLE 16 (Continued)

Io = 1.0E-0009, R = 0.907, %Tr = 30, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.3091	0.4633	0.6671	0.1543
10	0.2859	0.4389	0.6514	0.1530
20	0.2578	0.4088	0.6307	0.1510
30	0.2425	0.3917	0.6191	0.1492
40	0.2156	0.3634	0.5933	0.1478
50	0.2157	0.3648	0.5915	0.1490
60	0.1713	0.3174	0.5396	0.1461
70	0.1622	0.3072	0.5281	0.1450
80	0.1711	0.3164	0.5409	0.1453
90	0.1659	0.3117	0.5322	0.1458

Io = 1.0E-0007, R = 0.896, %Tr = 30, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.2359	0.3447	0.6843	0.1088
10	0.2182	0.3267	0.6679	0.1085
20	0.1968	0.3046	0.6462	0.1078
30	0.1851	0.2920	0.6340	0.1069
40	0.1646	0.2712	0.6070	0.1066
50	0.1647	0.2721	0.6052	0.1075
60	0.1308	0.2373	0.5512	0.1065
70	0.1239	0.2298	0.5392	0.1059
80	0.1307	0.2366	0.5524	0.1059
90	0.1267	0.2331	0.5435	0.1064

TABLE 16 (Continued)

Io = 1.0E-0005, R = 0.885, %Tr = 30, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.1619	0.2272	0.7127	0.0653
10	0.1498	0.2155	0.6954	0.0656
20	0.1352	0.2010	0.6725	0.0658
30	0.1272	0.1928	0.6596	0.0656
40	0.1131	0.1792	0.6313	0.0661
50	0.1132	0.1798	0.6295	0.0666
60	0.0900	0.1570	0.5730	0.0671
70	0.0853	0.1521	0.5605	0.0669
80	0.0899	0.1566	0.5742	0.0667
90	0.0872	0.1543	0.5652	0.0671

Io = 1.0E-0003, R = 0.882, %Tr = 30, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.0865	0.1129	0.7663	0.0264
10	0.0801	0.1070	0.7492	0.0268
20	0.0724	0.0996	0.7267	0.0272
30	0.0681	0.0954	0.7142	0.0273
40	0.0607	0.0884	0.6863	0.0277
50	0.0607	0.0887	0.6846	0.0280
60	0.0485	0.0771	0.6286	0.0286
70	0.0460	0.0746	0.6162	0.0286
80	0.0484	0.0769	0.6297	0.0285
90	0.0470	0.0757	0.6212	0.0287

TABLE 16 (Continued)

Io = 1.0E-0002, R = 0.900, %Tr = 30, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.0496	0.0605	0.8195	0.0109
10	0.0460	0.0570	0.8058	0.0111
20	0.0415	0.0527	0.7877	0.0112
30	0.0391	0.0503	0.7777	0.0112
40	0.0349	0.0463	0.7544	0.0114
50	0.0349	0.0464	0.7529	0.0115
60	0.0280	0.0396	0.7054	0.0117
70	0.0265	0.0382	0.6948	0.0117
80	0.0279	0.0395	0.7067	0.0116
90	0.0271	0.0388	0.6991	0.0117

TABLE 17

FIXED ARRAY, OPTIMIZED FOR η_E
$\%_{TR}$ = 30, $\%_{CLEAR}$ = 80

Io = 1.0E-0013, R = 1.092, %Tr = 30, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.5328	0.7023	0.7588	0.1694
10	0.4930	0.6646	0.7417	0.1717
20	0.4445	0.6182	0.7191	0.1736
30	0.4181	0.5918	0.7065	0.1737
40	0.3718	0.5483	0.6782	0.1765
50	0.3720	0.5503	0.6760	0.1783
60	0.2955	0.4773	0.6191	0.1818
70	0.2799	0.4616	0.6065	0.1817
80	0.2952	0.4758	0.6205	0.1806
90	0.2863	0.4686	0.6110	0.1823

Io = 1.0E-0011, R = 1.088, %Tr = 30, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.4457	0.5826	0.7650	0.1369
10	0.4123	0.5516	0.7475	0.1393
20	0.3719	0.5134	0.7244	0.1415
30	0.3498	0.4917	0.7115	0.1418
40	0.3111	0.4558	0.6825	0.1447
50	0.3113	0.4575	0.6804	0.1462
60	0.2473	0.3974	0.6224	0.1501
70	0.2343	0.3844	0.6096	0.1501
80	0.2471	0.3961	0.6238	0.1490
90	0.2397	0.3902	0.6143	0.1505

TABLE 17 (Continued)

Io = 1.0E-0009, R = 1.084, %Tr = 30, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.3583	0.4633	0.7733	0.1050
10	0.3315	0.4389	0.7552	0.1074
20	0.2990	0.4088	0.7315	0.1098
30	0.2813	0.3917	0.7181	0.1104
40	0.2502	0.3634	0.6885	0.1132
50	0.2504	0.3648	0.6864	0.1144
60	0.1990	0.3174	0.6271	0.1184
70	0.1886	0.3072	0.6141	0.1186
80	0.1988	0.3164	0.6284	0.1176
90	0.1930	0.3117	0.6190	0.1188

Io = 1.0E-0007, R = 1.078, %Tr = 30, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.2701	0.3447	0.7836	0.0746
10	0.2499	0.3267	0.7649	0.0768
20	0.2255	0.3046	0.7404	0.0791
30	0.2122	0.2920	0.7267	0.0798
40	0.1888	0.2712	0.6962	0.0824
50	0.1890	0.2721	0.6943	0.0832
60	0.1504	0.2373	0.6337	0.0869
70	0.1425	0.2298	0.6203	0.0872
80	0.1502	0.2366	0.6349	0.0864
90	0.1458	0.2331	0.6256	0.0873

TABLE 17 (Continued)

Io = 1.0E-0005, R = 1.070, %Tr = 30, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.1814	0.2272	0.7984	0.0458
10	0.1679	0.2155	0.7792	0.0476
20	0.1516	0.2010	0.7542	0.0494
30	0.1427	0.1928	0.7401	0.0501
40	0.1271	0.1792	0.7091	0.0521
50	0.1272	0.1798	0.7075	0.0526
60	0.1014	0.1570	0.6460	0.0556
70	0.0962	0.1521	0.6324	0.0559
80	0.1013	0.1566	0.6471	0.0553
90	0.0985	0.1543	0.6381	0.0559

Io = 1.0E-0003, R = 1.062, %Tr = 30, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.0933	0.1129	0.8264	0.0196
10	0.0865	0.1070	0.8087	0.0205
20	0.0782	0.0996	0.7854	0.0214
30	0.0737	0.0954	0.7724	0.0217
40	0.0657	0.0884	0.7435	0.0227
50	0.0658	0.0887	0.7421	0.0229
60	0.0528	0.0771	0.6844	0.0243
70	0.0501	0.0746	0.6715	0.0245
80	0.0527	0.0769	0.6854	0.0242
90	0.0513	0.0757	0.6772	0.0244

TABLE 17 (Continued)

Io = 1.0E-0002, R = 1.062, %Tr = 30, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.0520	0.0605	0.8595	0.0085
10	0.0482	0.0570	0.8459	0.0088
20	0.0436	0.0527	0.8279	0.0091
30	0.0411	0.0503	0.8180	0.0092
40	0.0368	0.0463	0.7948	0.0095
50	0.0368	0.0464	0.7935	0.0096
60	0.0296	0.0396	0.7464	0.0101
70	0.0281	0.0382	0.7358	0.0101
80	0.0295	0.0395	0.7475	0.0100
90	0.0288	0.0388	0.7404	0.0101

TABLE 18

TRACKING ARRAY
$\%_{TR} = 50$, $\%_{CLEAR} = 80$

Io = 1.0E-0013, R = 0.923, %Tr = 50, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.5898	0.6928	0.8514	0.1029
10	0.5456	0.6600	0.8267	0.1144
20	0.5014	0.6272	0.7994	0.1258
30	0.4572	0.5944	0.7691	0.1372
40	0.4129	0.5616	0.7353	0.1487
50	0.3687	0.5288	0.6972	0.1601
60	0.3245	0.4960	0.6542	0.1715
70	0.2803	0.4633	0.6050	0.1830
80	0.2360	0.4305	0.5483	0.1944
90	0.1918	0.3977	0.4824	0.2059

Io = 1.0E-0011, R = 0.915, %Tr = 50, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.4944	0.5734	0.8621	0.0790
10	0.4573	0.5464	0.8370	0.0891
20	0.4203	0.5194	0.8092	0.0991
30	0.3832	0.4923	0.7784	0.1091
40	0.3462	0.4653	0.7440	0.1191
50	0.3091	0.4383	0.7053	0.1291
60	0.2721	0.4113	0.6616	0.1392
70	0.2351	0.3842	0.6118	0.1492
80	0.1980	0.3572	0.5543	0.1592
90	0.1610	0.3302	0.4875	0.1692

TABLE 18 (Continued)

Io = 1.0E-0009, R = 0.907, %Tr = 50, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.3986	0.4549	0.8762	0.0563
10	0.3687	0.4335	0.8506	0.0648
20	0.3389	0.4122	0.8223	0.0732
30	0.3091	0.3908	0.7909	0.0817
40	0.2793	0.3695	0.7559	0.0902
50	0.2495	0.3481	0.7166	0.0986
60	0.2197	0.3268	0.6722	0.1071
70	0.1899	0.3054	0.6216	0.1156
80	0.1601	0.2841	0.5634	0.1240
90	0.1302	0.2628	0.4957	0.1325

Io = 1.0E-0007, R = 0.896, %Tr = 50, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.3019	0.3376	0.8941	0.0357
10	0.2794	0.3218	0.8682	0.0424
20	0.2569	0.3060	0.8396	0.0491
30	0.2345	0.2902	0.8079	0.0557
40	0.2120	0.2744	0.7725	0.0624
50	0.1895	0.2586	0.7329	0.0691
60	0.1671	0.2428	0.6880	0.0758
70	0.1446	0.2270	0.6369	0.0824
80	0.1221	0.2112	0.5782	0.0891
90	0.0997	0.1955	0.5100	0.0958

TABLE 18 (Continued)

Io = 1.0E-0005, R = 0.885, %Tr = 50, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.2037	0.2226	0.9150	0.0189
10	0.1888	0.2122	0.8897	0.0234
20	0.1738	0.2017	0.8618	0.0279
30	0.1589	0.1912	0.8308	0.0324
40	0.1439	0.1808	0.7962	0.0368
50	0.1290	0.1703	0.7574	0.0413
60	0.1141	0.1599	0.7135	0.0458
70	0.0991	0.1494	0.6635	0.0503
80	0.0842	0.1390	0.6059	0.0548
90	0.0693	0.1285	0.5390	0.0592

Io = 1.0E-0003, R = 0.882, %Tr = 50, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.1057	0.1127	0.9377	0.0070
10	0.0982	0.1072	0.9163	0.0090
20	0.0907	0.1016	0.8926	0.0109
30	0.0833	0.0961	0.8662	0.0129
40	0.0758	0.0906	0.8366	0.0148
50	0.0683	0.0851	0.8031	0.0167
60	0.0609	0.0796	0.7650	0.0187
70	0.0534	0.0740	0.7212	0.0206
80	0.0459	0.0685	0.6704	0.0226
90	0.0385	0.0630	0.6106	0.0245

TABLE 18 (Continued)

Io = 1.0E-0002, R = 0.900, %Tr = 50, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.0599	0.0629	0.9517	0.0030
10	0.0557	0.0596	0.9354	0.0038
20	0.0516	0.0562	0.9172	0.0047
30	0.0474	0.0529	0.8967	0.0055
40	0.0433	0.0496	0.8735	0.0063
50	0.0392	0.0462	0.8468	0.0071
60	0.0350	0.0429	0.8161	0.0079
70	0.0309	0.0396	0.7801	0.0087
80	0.0267	0.0362	0.7375	0.0095
90	0.0226	0.0329	0.6863	0.0103

TABLE 19

FIXED ARRAY, OPTIMIZED FOR POWER
$\%_{TR}$ = 50, $\%_{CLEAR}$ = 80

Io = 1.0E-0013, R = 0.923, %Tr = 50, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.4570	0.7048	0.6484	0.2478
10	0.4283	0.6782	0.6315	0.2499
20	0.3944	0.6451	0.6114	0.2507
30	0.3753	0.6272	0.5984	0.2519
40	0.3456	0.5985	0.5774	0.2529
50	0.3459	0.5984	0.5781	0.2525
60	0.2894	0.5452	0.5308	0.2558
70	0.2778	0.5349	0.5194	0.2571
80	0.2859	0.5408	0.5288	0.2548
90	0.2840	0.5390	0.5269	0.2550

Io = 1.0E-0011, R = 0.915, %Tr = 50, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.3834	0.5847	0.6558	0.2013
10	0.3594	0.5628	0.6386	0.2034
20	0.3310	0.5356	0.6180	0.2046
30	0.3149	0.5207	0.6047	0.2058
40	0.2900	0.4972	0.5833	0.2072
50	0.2903	0.4971	0.5840	0.2068
60	0.2428	0.4532	0.5358	0.2104
70	0.2332	0.4448	0.5242	0.2116
80	0.2399	0.4496	0.5337	0.2096
90	0.2383	0.4481	0.5318	0.2098

TABLE 19 (Continued)

Io = 1.0E-0009, R = 0.907, %Tr = 50, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.3100	0.4650	0.6668	0.1549
10	0.2906	0.4477	0.6491	0.1571
20	0.2677	0.4262	0.6280	0.1586
30	0.2547	0.4146	0.6144	0.1599
40	0.2345	0.3960	0.5923	0.1614
50	0.2348	0.3959	0.5930	0.1611
60	0.1964	0.3613	0.5437	0.1649
70	0.1886	0.3547	0.5319	0.1660
80	0.1941	0.3585	0.5415	0.1644
90	0.1928	0.3573	0.5395	0.1645

Io = 1.0E-0007, R = 0.896, %Tr = 50, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.2366	0.3459	0.6841	0.1093
10	0.2218	0.3331	0.6658	0.1113
20	0.2044	0.3173	0.6440	0.1130
30	0.1945	0.3087	0.6300	0.1142
40	0.1791	0.2950	0.6072	0.1159
50	0.1793	0.2950	0.6079	0.1157
60	0.1501	0.2695	0.5571	0.1193
70	0.1442	0.2646	0.5451	0.1203
80	0.1484	0.2674	0.5548	0.1190
90	0.1473	0.2665	0.5529	0.1192

TABLE 19 (Continued)

Io = 1.0E-0005, R = 0.885, %Tr = 50, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.1625	0.2280	0.7126	0.0655
10	0.1524	0.2196	0.6938	0.0672
20	0.1405	0.2093	0.6715	0.0687
30	0.1338	0.2036	0.6571	0.0698
40	0.1233	0.1946	0.6338	0.0713
50	0.1234	0.1946	0.6344	0.0711
60	0.1036	0.1778	0.5825	0.0742
70	0.0996	0.1746	0.5704	0.0750
80	0.1024	0.1765	0.5800	0.0741
90	0.1017	0.1759	0.5780	0.0742

Io = 1.0E-0003, R = 0.882, %Tr = 50, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.0868	0.1133	0.7663	0.0265
10	0.0816	0.1090	0.7484	0.0274
20	0.0754	0.1037	0.7277	0.0282
30	0.0719	0.1007	0.7141	0.0288
40	0.0665	0.0961	0.6919	0.0296
50	0.0665	0.0961	0.6925	0.0295
60	0.0562	0.0874	0.6431	0.0312
70	0.0542	0.0857	0.6319	0.0316
80	0.0556	0.0867	0.6406	0.0312
90	0.0552	0.0864	0.6387	0.0312

TABLE 19 (Continued)

Io = 1.0E-0002, R = 0.900, %Tr = 50, %Clr = 80

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.0498	0.0607	0.8194	0.0110
10	0.0468	0.0582	0.8046	0.0114
20	0.0434	0.0550	0.7878	0.0117
30	0.0414	0.0533	0.7765	0.0119
40	0.0383	0.0506	0.7579	0.0122
50	0.0384	0.0506	0.7584	0.0122
60	0.0326	0.0455	0.7163	0.0129
70	0.0314	0.0445	0.7067	0.0130
80	0.0322	0.0451	0.7142	0.0129
90	0.0320	0.0449	0.7125	0.0129

TABLE 20

FIXED ARRAY, OPTIMIZED FOR η_E
$\%_{TR}$ = 50, $\%_{CLEAR}$ = 80

Io = 1.0E-0013, R = 1.092, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.7282	0.8678	0.8391	0.1396
10	0.6847	0.8342	0.8209	0.1494
20	0.6331	0.7929	0.7985	0.1597
30	0.5985	0.7662	0.7812	0.1677
40	0.5606	0.7360	0.7617	0.1754
50	0.5599	0.7354	0.7613	0.1755
60	0.4759	0.6695	0.7109	0.1935
70	0.4659	0.6617	0.7041	0.1958
80	0.4653	0.6611	0.7038	0.1958
90	0.4621	0.6587	0.7015	0.1966

Io = 1.0E-0011, R = 1.088, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.6061	0.7192	0.8427	0.1132
10	0.5700	0.6915	0.8242	0.1215
20	0.5274	0.6575	0.8022	0.1300
30	0.4988	0.6355	0.7849	0.1367
40	0.4674	0.6106	0.7654	0.1432
50	0.4667	0.6100	0.7651	0.1433
60	0.3973	0.5557	0.7149	0.1585
70	0.3889	0.5494	0.7080	0.1604
80	0.3885	0.5488	0.7079	0.1603
90	0.3858	0.5469	0.7055	0.1611

TABLE 20 (Continued)

Io = 1.0E-0009, R = 1.084, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.4843	0.5715	0.8474	0.0872
10	0.4555	0.5496	0.8289	0.0940
20	0.4220	0.5227	0.8074	0.1007
30	0.3993	0.5053	0.7902	0.1060
40	0.3744	0.4856	0.7710	0.1112
50	0.3739	0.4852	0.7705	0.1113
60	0.3189	0.4423	0.7209	0.1234
70	0.3123	0.4373	0.7141	0.1250
80	0.3120	0.4368	0.7142	0.1249
90	0.3098	0.4353	0.7117	0.1255

Io = 1.0E-0007, R = 1.078, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.3630	0.4250	0.8542	0.0620
10	0.3417	0.4088	0.8359	0.0671
20	0.3170	0.3889	0.8151	0.0719
30	0.3002	0.3760	0.7983	0.0758
40	0.2818	0.3615	0.7796	0.0797
50	0.2814	0.3612	0.7791	0.0798
60	0.2407	0.3294	0.7307	0.0887
70	0.2358	0.3257	0.7241	0.0899
80	0.2356	0.3253	0.7243	0.0897
90	0.2340	0.3242	0.7217	0.0902

TABLE 20 (Continued)

Io = 1.0E-0005, R = 1.070, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.2428	0.2807	0.8651	0.0378
10	0.2289	0.2700	0.8477	0.0411
20	0.2127	0.2568	0.8284	0.0441
30	0.2018	0.2483	0.8126	0.0465
40	0.1898	0.2387	0.7950	0.0489
50	0.1895	0.2385	0.7945	0.0490
60	0.1629	0.2175	0.7490	0.0546
70	0.1597	0.2150	0.7429	0.0553
80	0.1596	0.2148	0.7432	0.0551
90	0.1585	0.2140	0.7406	0.0555

Io = 1.0E-0003, R = 1.062, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.1257	0.1416	0.8878	0.0159
10	0.1187	0.1359	0.8731	0.0173
20	0.1106	0.1290	0.8571	0.0184
30	0.1051	0.1245	0.8438	0.0195
40	0.0990	0.1195	0.8289	0.0204
50	0.0989	0.1194	0.8283	0.0205
60	0.0855	0.1083	0.7896	0.0228
70	0.0840	0.1070	0.7846	0.0231
80	0.0839	0.1068	0.7848	0.0230
90	0.0833	0.1065	0.7824	0.0232

TABLE 20 (Continued)

Io = 1.0E-0002, R = 1.062, %Tr = 50, %Clr = 100

%Cov	E_N	E_{NMAX}	η_E	Loss
0	0.0713	0.0781	0.9128	0.0068
10	0.0674	0.0747	0.9014	0.0074
20	0.0627	0.0706	0.8889	0.0078
30	0.0596	0.0678	0.8783	0.0083
40	0.0561	0.0648	0.8664	0.0087
50	0.0561	0.0647	0.8660	0.0087
60	0.0484	0.0580	0.8346	0.0096
70	0.0476	0.0573	0.8305	0.0097
80	0.0475	0.0572	0.8306	0.0097
90	0.0472	0.0569	0.8285	0.0098

APPENDIX B

SIMULATION PROGRAM

```
program Solar;
{$M 10240,0,655360}
{$I-}
{$DEFINE BigProg }
uses CRT,SFunct,FileIO;


{

        This is the main program, which displays the menu and calls the proper
        procedures.


}

var

        Choice                                          :char;
        Continue                                        :boolean;
        CommandFile,OutputFile                  :string;



procedure CommFile;

begin

        if FileRead then
         begin
                writeln;
                write('Command File Name:  ');
                readln(CommandFile);
                assign(W,CommandFile);
                reset(W);
                if ErrOut(CommandFile) then
                        FileRead := FALSE;
         end;
end;


begin

        TextMode(CO80 + Font8x8);
        Window(5,10,Lo(WindMax),Hi(WindMax));

        assign(T,'\solar\RngName');

        Choice := 'A';
        Continue := TRUE;
        FileRead := FALSE;
        if (ParamStr(1) <> '')then
                FileRead := TRUE
         else
```

```
        FileRead := FALSE;

CommFile;


while Continue do
 begin
        DirectVideo := False;
        ClrScr;

        writeln('1.   Generate day(s)');
        writeln('2.   Plot day');
        writeln('3.   Analyze circuit');
        writeln('4.   Calculate Maximum Power Points');
        writeln('5.   Plot circuit output');
        writeln('6.   Change calculation intervals');
        writeln('7.   Run Using Command File');
        writeln('8.   Output Name for Batch');
        writeln('9.   Convert .PPU to .PRT');
        writeln;

        if FileRead then
                if Eof(W) then
                        FileRead := FALSE;

        if FileRead then
                read(W,Choice)
         else
                readln(Choice);

        Choice := UpCase(Choice);

        case Choice of

                '1': begin
                                write('Automatic Filenames?  ');
                                if FileRead then
                                        read(W,Choice)
                                 else
                                        readln(Choice);
                                Choice := UpCase(Choice);
                                if (Choice = 'Y') then

                                        DayGen('X')
                                 else
                                        DayGen('R');
                        end;

                '2': PlotRoutine;

                '3': Calculate;
```

```
                    '4': MPPoint;

                    '5': PlotCirc;

                    '6': ChangeSetup;

                    '7': begin
                                FileRead := TRUE;
                                CommFile;
                         end;

                    '8': begin
                                rewrite(T);
                                writeln;
                                write('Output File Name:  ');
                                if FileRead then
                                 begin
                                        read(W,OutPutFile);
                                        writeln(OutPutFile);
                                 end
                                else
                                        readln(OutPutFile);

                                writeln(T,OutPutFile);
                                close(T);
                         end;


                    '9': begin
                                FileName := '';
                                MakePRT;
                         end;

                    'X': Continue := FALSE;
               end;
          end;
end.
```

```
unit SFunct;
{$I-}
{$DEFINE BigProg}
interface

uses BGIUnit,FileIO;


{ Let the main program know where the procedures are. }

procedure DayGen(NameFlag: char);
procedure Calculate;
procedure PlotRoutine;
procedure PlotCirc;
procedure MPPoint;
procedure ChangeSetup;
procedure MakePRT;
type

     TestRange =     record
                         Start,Value,Stop,Increment: integer;
                    end;



var

        A,F,G,W,T                                          :text;

        i,j,Ins,NumPts,Code                               :integer;
        NumbPoints,GraphDriver,GraphMode                   :integer;
        X,Y,YPos,XPos,MaxVal,DummyInt                        :integer;
        NumVal,DayInterval                                  :integer;

        TrInt,PerCoInt,PerClInt                           :TestRange;

        FileName,OutFile,DataName,OutStr,Value             :string;
        DataLabel, Param,LastDataFile,LastPPUFile          :string;

        Velocity                                          :single;
        EN,ENmax,Eff,Pmp,Ppu,Pmpu,Loss                         :single;
        EBase,VBase,Temp                                    :single;

        Key,Condition,CloudSize                             :char;
        Junk                                              :string[7];

        XReal,YReal,DummyReal                                 :single;
        Transmitt,PerCover,PerClear,Insr,Cloud,m,Z,E :single;

        XPosL,YPosL                                         :longint;

        Io,Imax,VmpuPrime,Result,XBar,Xr,Yr                :extended;
        lnXBar                                             :extended;
```

```
        Vpu,IpuPrime,Ipu,Vmpu,Impu,Ri,Rf,Rpu,RInc     :extended;
        R,Vout,Vmout                                  :extended;

        Is                                            :array [1..4320] of single;

        Done,IntVar,AppendFile,FirstTime,UserName     :boolean;
        RangeData,FirstPlot,Tracking                  :boolean;
        FileRead,SaveValue                            :boolean;


{       Variable Definitions

                Io        :Dark Current, Amperes
                Is        :Short Circuit Current, p.u.
                Ipu       :Load Current, p.u.
                Impu      :Load Current at Maximum Power Point, p.u.
                Vpu       :Load Voltage, p.u.
                Vmpu      :Load Voltage at Maximum Power Point, p.u.
                Ppu       :Load Power, p.u.
                Pmpu      :Maximum Power, p.u.
                EBase     :Base value for energy
                EN        :Total Energy, p.u. x minutes
                ENmax     :Maximum Total Energy, p.u. x minutes
                Eff       :Utilization Efficiency = EN/ENmax
                R         :Load Resistance, ohms
                Rpu       :Load Resistance, per-unit
                Temp      :Temperature
                TrInt     :Percent of cloud transmittance
                PerCoInt  :Percent cover
                PerClInt  :Percent clear
}


implementation


uses DOS,CRT,Graph,XDump,Strings;

var

        H,L                     :text;

        PrnLArea,PrnUArea       : rect;
        PSR                     : PSptr;  {pointer variable used to access PSrec}


{ This function raises a real number to a real number power. }

function XY(X,Y: extended): extended;
```

```
{ Computes X ^ Y }
begin
     if (X > 0) then
          XY := exp(Y * ln(X))
     else if (X = 0) then
          XY := 0
     else if (X < 0) then
          XY := -exp(Y * ln(-X));
end;




{    This function is used by the GetFileName procedure when the
     program is to generate/use days over a range of transmittance,
     clear, or cover values.
}
procedure InputRange(DataLabel: string; var GetRange: TestRange);
begin
        with GetRange do
         begin
                writeln(DataLabel,' Values: ');
                write('                 Starting Value: ');
                if FileRead then
                 begin
                        read(W,Start);
                        writeln(Start);
                 end
                else
                        readln(Start);
                write('                 Ending value:    ');

                if FileRead then
                 begin
                        read(W,Stop);
                        writeln(Stop);
                 end
                else
                        readln(Stop);
                write('                 Increment:       ');

                if FileRead then
                 begin
                        read(W,Increment);
                        writeln(Increment);
                 end
                else
                        readln(Increment);
                MaxVal := Stop;
                Value := Start;
         end;
```

```
end;



{    This procedure is used when the simulation/day generation is
     to be done for only a single value of one of the parameters.
}
Procedure SingleInput(DataLabel: string; var GetRange:
TestRange);
begin
        with GetRange do
          begin
                write(DataLabel,':       ');
                if FileRead then
                 begin
                   read(W,Value);
                   writeln(Value);
                 end
                else
                   readln(Value);
                Start := Value;
                Stop  := Value;
                Increment := 1;
          end;
end;



{    This function returns the proper filename for the data
        file corresponding to the given transmittance, clear,
        and cover values.
}
function GetFileName(FileName :string; Range: boolean):string;

var
        Directory            : string;
        Choice               : char;


begin

        if FirstTime then
         begin

                writeln;
                write('(T)racking or (F)ixed  ');
                if FileRead then
                 begin
                        read(W,Choice);
```

```
        writeln(Choice);
 end
else
        readln(Choice);
if (Choice = 'T') or (Choice = 't') then
        Tracking := TRUE
 else
        Tracking := FALSE;
if not Tracking then
 begin
        write('Cloud Size: (S)mall, (M)edium, (L)arge');
        if FileRead then
         begin
                read(W,CloudSize);
                writeln(CloudSize);
         end
        else
                readln(CloudSize);
        writeln;
        CloudSize := UpCase(CloudSize);
        writeln;
        write('Cloud velocity (m/s)   ');
        if FileRead then
         begin
                read(W,Velocity);
                writeln(Velocity:5:1);
         end
        else
                readln(Velocity);
end;

Choice := 'N';

if Range then
 begin
        writeln;
        write ('Vary data parameters?    ');
        if FileRead then
         begin
                read(W,Choice);
                if (Choice = ' ') then
                        read(W,Choice);
                writeln(Choice);
         end
        else
                readln(Choice);
        Choice := UpCase(Choice);
 end;

if (Choice = 'Y') then
 begin
        writeln;
        write('(T)ransmittance, % C(O)ver, % C(L)ear,(A)ll:  ');
```

```
            if FileRead then
             begin
                    read(W,Choice);
                    writeln(Choice);
             end
            else
                    readln(Choice);
            Choice := UpCase(Choice);
            Condition := Choice;
      end
      else
       begin
            Condition := 'N';
            MaxVal    := 100;
       end;

     if (Choice <> 'A') then
      begin
              writeln;
              if ( Choice = 'T') then
                           InputRange('Transmittance',TrInt)
               else
                           SingleInput('Transmittance',TrInt);

        writeln;
              if ( Choice = 'O') then
                           InputRange('Percentage Cover',PerCoInt)
               else
                           SingleInput('Percentage Cover',PerCoInt);

              writeln;
              if ( Choice = 'L') then
                           InputRange('Percentage Clear',PerClInt)
               else
                           SingleInput('Percentage Clear',PerClInt);
     end;
     writeln;
     if (Choice = 'A') then
      begin
            InputRange('Transmittance',TrInt);
            InputRange('Percentage Cover',PerCoInt);
            InputRange('Percentage Clear',PerClInt);
      end;


     FirstTime := FALSE;
 end;

FileName := UpperCase(FileName);

if (FileName = 'R') then
 begin
     write('Output File Name: ');
```

```
        if FileRead then
         begin
               read(W,FileName);
               writeln(FileName);
         end
        else
               readln(FileName);
end


else
 begin
        with Trint do

               if (Value = 100) then
                       FileName := 'A0'
               else
               if (Value < 10) then
                begin
                       Str(Value:1,FileName);
                       FileName := '0'+FileName;
                end
               else
                       Str(Value:2,FileName);

        with PerCoInt do

               if (Value = 100) then
                       Param := 'A0'
               else
               if (Value < 10) then
                begin
                       Str(Value:1,Param);
                       Param := '0' + Param;
                end
               else
                       Str(Value:2,Param);

        FileName := FileName + Param;


        with PerClInt do

               if (Value = 100) then
                       Param := 'A0'
               else
               if (Value < 10) then
               begin
                       Str(Value:1,Param);
                       Param := '0' + Param;
               end
               else
```

```
                        Str(Value:2,Param);

                FileName := FileName + Param;
        end;

        if Tracking then
                FileName := 'T' + FileName;

        if (Pos('.',FileName) = 0) then
                FileName := FileName + '.DAT';

        GetDir(0,Directory);
        GetFileName := Directory + '\' + FileName;
end;




{   This procedure plots a file of solar data.
}
procedure PlotData(FileName: string; var Quit: boolean);

var

        AllOK           : boolean;
        Choice          : char;
        Scale           : real;

begin


        Done := FALSE;
        GraphMode := 2;
        GraphDriver := 9;


        while (not Done) do
        begin

        while (FileName = '') do
         begin
                ClrScr;
                write('Data File: ');
                readln(FileName);
        end;

        if (Pos('.',FileName) = 0) then
                FileName := FileName + '.DAT';

        assign(F,FileName);
        reset(F);
```

```
if not ErrOut(FileName) then
 begin
        while (not Done) do
         begin
               readln(F,DataName);
               readln(F,NumbPoints,DummyReal,DummyReal);
               Scale := NumbPoints/550;
               AllOK := errout(FileName);
               if FirstPlot then
                begin
                       InitGraph(GraphDriver,GraphMode,'');
                       FirstPlot := FALSE;
                end;
               DummyReal := NumbPoints/(10.0);
               MaxVal := Trunc(DummyReal) + 1;
               SetLineStyle(SolidLn,0,NormWidth);
               Line(50,50,50,405);
               Line(45,400,600,400);

               for Y := 0 to 9 do
                begin
                       YPos := Y*35 + 50;
                       Line(45,YPos,55,YPos);
               end;

               for X := 0 to 10 do
                begin
                       XPos := X*55 + 50;
                       Line(XPos,395,XPos,405);
               end;

               SetTextStyle(2,0,6);
               SetTextJustify(CenterText,CenterText);

               for Y := 0 to 10 do
                begin
                       YPos := 400 - Y*35;
                       MoveTo(25,YPos);
                       YReal := Y/10.0;
                       Str(YReal:3:1,OutStr);
                       OutText(OutStr);
               end;

               for X := 0 to 10 do
                begin
                       XPos := 50 + X*55;
                       MoveTo(XPos,415);
                       DummyInt := round(X*NumbPoints/10);
                       Str(DummyInt,OutStr);
                       OutText(OutStr);
               end;

               readln(F,Value);
```

```
        if (Pos('.',Value) = 0) then
        begin
            Val('Value',Y,Code);
            IntVar := TRUE;
        end
        else
        begin
            Val('Value',DummyReal,Code);
            Y := Round(100*DummyReal);
            IntVar := FALSE;
        end;

        YPos := 400 - Y;
        MoveTo(50,YPos);
        for i := 2 to NumbPoints do
        begin
                if IntVar then
                    readln(F,Y)
                else
                begin
                    readln(F,DummyReal);
                    Y := Round(DummyReal*100);
                end;
                YPos := 400 - Round(3.5*Y);
                XPos := 50 + round(i/NumbPoints*550);
                LineTo(XPos,YPos);
        end;

        readln(F,Junk);
        readln(F,Junk,YReal);
        YPos := 400 - Round(3.5*YReal);
        Line(50,YPos,600,YPos);
        MoveTo(320,440);
        OutText(DataName);

        Done := TRUE;
end;
close(F);
Done := False;
while (not Done) do
begin
    Key := ReadKey;

    case Key of

        #13: begin
                Done := True;
             end;
        #16: begin
                Done := True;
                Quit := True;
             end;
        #25: PScreen(PSR^)
```

```
                    end;

                 end;
             end
             else
              begin
                      writeln('Strike Any Key to Continue...');
                      Choice := ReadKey;
                      Done    := True;
              end;
          end;
          ClearViewPort;
end;



{     This procedure plots the circuit outputs
}
procedure PlotCirc;

type
     InputData = record
                    PerTr,PerCl,PerCo,R,ENENmaxu,Eff,Loss,PlotX,PlotY:real;
               end;

var

          AllOK                : boolean;
          Choice               : char;
          Data                 : array [1..100] of InputData;
          Dummy                : string[5];
          i                    : integer;
          Value,MaxY           : single;
          YLabel               : string;
          FirstData            : Boolean;
          DummyReal            : single;
begin

           FirstData := TRUE;
           Done := FALSE;
           GraphMode := 2;
           GraphDriver := 9;
           ClrScr;
           write('Data File: ');
           readln(FileName);
           if ((FileName = 'L') or (FileName = 'l')) then
            begin
                  reset(H);
                  readln(H,FileName);
                  readln(H,FileName);
                  close(H);
```

```
 end;
if (Pos('.',FileName) = 0) then
       FileName := FileName + '.ppu';

assign(F,FileName);
reset(F);

if not ErrOut(FileName) then
 begin
        ClrScr;
        write('Plot (E)pu, E(m)pu, E(f)f, or (L)oss  ');
        readln(Choice);
        Choice := UpCase(Choice);
        case Choice of
                'E': YLabel := 'Energy Output';
                'M': YLabel := 'Maximum Available Energy';
                'F': YLabel := 'Utilization Efficiency';
                'L': YLabel := 'Energy Loss';
         end;

        InitGraph(GraphDriver,GraphMode,'');
        SetLineStyle(SolidLn,0,NormWidth);
        Line(85,50,85,405);
        Line(80,400,630,400);

        for Y := 0 to 9 do
         begin
                YPos := Y*35 + 50;
                Line(80,YPos,90,YPos);
        end;

        while (not Eof(F)) do
         begin
               readln(F,DataName);
               readln(F,DataName);
               readln(F,Condition,MaxVal);
               case Condition of
                'T': DataName := '% Cloud Transmittance';
                'L': DataName := '% of Maximum Clear Sky';
                'O': DataName := '% Cloud Cover';
                end;
               readln(F,Dummy,Io);
               for i := 1 to 3 do
                       readln(F,Param);

               AllOK  := errout(FileName);

               for X := 0 to (MaxVal div 10) do
                begin
                       XPos := trunc(5450/MaxVal*X) + 85;
                       Line(XPos,395,XPos,405);
                end;
```

```
SetTextStyle(2,0,6);
SetTextJustify(CenterText,CenterText);

for X := 0 to (MaxVal div 10) do
 begin
        XPos := 85 + trunc(5450/MaxVal*X);
        MoveTo(XPos,415);
        DummyInt := X*10;
        Str(DummyInt,OutStr);
        OutText(OutStr);
 end;
MaxY  := 0.0;
Value := 0.0;
i := 0;
while Value < MaxVal do
 begin
        Inc(i);
        with Data[i] do
         begin
                readln(F,PerTr,PerCo,PerCl,R,EN,ENmax,Eff,Loss);
                case Choice of
                        'E': PlotY := EN;
                        'M': PlotY := ENmax;
                        'F': PlotY := Eff;
                        'L': PlotY := Loss;
                 end;

                case Condition of
                        'T': PlotX := PerTr;
                        'L': PlotX := PerCl;
                        'O': PlotX := PerCo;
                 end;
                Value := PlotX;
         end;
 end;
NumVal := i;
case Choice of
        'E': Maxy := 1.0;
        'M': Maxy := 1.0;
        'F': Maxy := 1.0;
        'L': Maxy := 1.0;
 end;

MoveTo(55,400);
OutText('0.00');
for Y := 1 to 10 do
 begin
        YPos := 400 - Y*35;
        MoveTo(55,YPos);
        YReal := Y*MaxY/10;
        Str(YReal:4:2,OutStr);
        OutText(OutStr);
end;
```

```
                Y := round(100*Data[1].PlotY/MaxY);
                IntVar := FALSE;

                YPos := 400 - round(3.5*Y);
                DummyReal := 545/MaxVal;
                MoveTo(85+trunc(Data[1].PlotX*DummyReal),YPos);

                for i := 2 to NumVal do
                begin
                        Y := round(100*Data[i].PlotY/MaxY);
                        YPos := 400 - round(3.5*Y);
                        XPosL := 85 + trunc(Data[i].PlotX*DummyReal);
                        LineTo(XPosL,YPos);
                end;
                MoveTo(320,440);
                OutText(DataName);
                MoveTo(2,240);
                SetTextStyle(2,1,6);
                OutText(YLabel);

                Done := TRUE;
        end;
        close(F);
        Done := False;
        while (not Done) do
        begin
            Key := ReadKey;

            case Key of

                #13: begin
                          Done := True;
                        end;
                #16: begin
                          Done := True;
                        end;
                #25: PScreen(PSR^)
            end;

        end;
   end
  else
    begin
                writeln('Strike Any Key to Continue...');
                Choice := ReadKey;
                Done    := True;
    end;

   CloseGraph;
{    TextMode(LastMode);}
end;
```

```
{    This procedure does the circuit analysis using a file of
        insolation data.
}
procedure MaxPU(FileName:string;  var Prompt:boolean);


var

        AllOK              :boolean;
        Choice,Space       :char;
        DayInt,DotLoc      :integer;
        DummyStr           :string;
        OutPutFile         :string;
        fI,fIprime         :extended;
        k1,k2,k3,VMul      :extended;
        Voc,Vmp            :extended;

begin


        if (FileName = '') then
        begin
            write('Enter Solar Data File Name:  ');
            readln(FileName);
        end;
        if (pos('.',FileName) <> 0) then
                delete(FileName,pos('.',FileName)+4,10)
        else
                FileName := FileName + '.DAT';
        OutPutFile := FileName;
        delete(OutPutFile,pos('.',OutPutFile)+1,3);
        OutPutFile := OutPutFile + 'CIR';
        assign(A,OutPutFile);
        rewrite(A);
        assign(F,FileName);
        reset(F);
        AllOK := not errout(FileName);
        if not AllOK then
         begin
                writeln('Strike Any Key to Continue...');
                Key := ReadKey;
                exit;
         end;
        readln(F,DataLabel);
        writeln(A,DataLabel);
        readln(F,NumVal,DayInt);
        writeln(A,NumVal,' ',DayInt);

        if Prompt then
        begin
```

```
        write('Output File Name:  ');
        if FileRead then
         begin
                read(W,Space);
                read(W,OutFile);
                writeln(OutFile);
         end
        else
                readln(OutFile);
end;

if (OutFile = '') and not UserName then
        OutFile  := FileName
 else
        UserName := TRUE;
if (pos('.',OutFile) <> 0) then
        delete(OutFile,pos('.',OutFile),6);
if RangeData then
 begin
        OutFile := OutFile + '.ppu';
        if Prompt then
         begin
                OutFile := FExpand(OutFile);
                reset(H);
                readln(H,DummyStr);
                close(H);
                rewrite(H);
                writeln(H,DummyStr);
                writeln(H,OutFile);
                close(H);
         end
 end
else
        OutFile := OutFile + '.rpu';

FileName := UpperCase(FileName);

if (Copy(FileName,pos('.',FileName) - 7,1) = 'T') then
        EBase := 1.0 * NumVal * DayInt
 else
        EBase := 2.0 * NumVal * DayInt / Pi;

assign(G,OutFile);

if UserName then
  if Prompt then
        if FSearch(OutFile,'c:\') <> '' then
         begin
                writeln;
                if FileRead then
                 begin
                        writeln('Output File Exists, Appending');
                        Choice := 'A';
```

```
                end
               else
                begin
               writeln('Output File Exists: (O)verwrite or (A)ppend  ');
                       Choice := Readkey;
                 end;
               if ((Choice = 'O') or (Choice = 'o')) then
                       AppendFile := FALSE
                else
                       AppendFile := TRUE;
        end
       else
               AppendFile := FALSE
  else
       AppendFile := TRUE
else
       AppendFile := FALSE;

if AppendFile then
       append(G)
 else
       rewrite(G);
writeln;
writeln('Calculating...');
AllOK := not errout(OutFile);
if not AllOK then
 begin
       writeln('Strike Any Key to Continue...');
       Key := ReadKey;
       exit;
 end;
if RangeData then
       DataLabel := 'Circuit Output, Varying input conditions'
else
 begin
       Delete(DataLabel,1,10);
       DataLabel := 'Circuit Output'+ DataLabel;
 end;
if AppendFile and not RangeData then
begin
       writeln(G,'');
       writeln(G,'-------------------------------------------');
       writeln(G,'');
end;
if (not RangeData) or Prompt then
 begin
       if RangeData then
               writeln(G);
       writeln(G,DataLabel);
       if RangeData then
               writeln(G,Condition,' ',MaxVal)
        else
               writeln(G,'');
```

```
        writeln(G,'Io = ',Io:6);
        writeln(G,'');
 end;
if RangeData and Prompt then
        write(G,'%Tr  %Cov %Clr ');
if (not RangeData) or Prompt then
 begin
        writeln(G,'  R           EN          ENmax          Eff         Loss');
        writeln(G,'');
 end;
Prompt := FALSE;
IMAX := 0;
for I := 1 to NumVal do
        readln(F,Is[I]);

close(F);

VBase := 1.38049E-23 * Temp / 1.60219E-19* ln(1 + 1.0E13);

Voc   := 1.38049E-23 * Temp / 1.60219E-19* ln(1 + 1/Io);

if (RInc = 0.0) then
        RInc := 0.1;
Rpu  := Ri;
while (Rpu <= Rf) do
begin

        R     := Rpu*Voc;
        VMul := 1.60219E-19 * R / (Temp * 1.38049E-23);

{ Calculate the load voltage corresponding to each value of Is }

        EN  := 0;
        ENmax := 0;
        Vmpu := 0.5;

        for I := 1 to NumVal do
        begin

                IpuPrime  := Is[I];
                Ipu        := 0.0;

                if IpuPrime > 0.0 then
                 begin
                        while (abs(Ipu - IpuPrime) > 1.0e-6) do
                         begin
                                Ipu  := IpuPrime;
                                k2   := exp(VMul*Ipu);
                                fI   := Is[I] - Io * (k2 - 1) - Ipu;
                                fIPrime   := -Io * VMul * k2  - 1;
                                IpuPrime  := Ipu - fI/fIPrime
                         end;
                        Ipu  := IpuPrime;
```

```
                Vout := Ipu * Rpu * Voc;
                Vpu  := Vout/VBase;

                xr   := Is[I];
                k1   := Is[I]/Io;
                yr   := ln(Is[I])/ln(Io);


                VmpuPrime := 1.0;
                while (abs(Vmpu - VmpuPrime) > 1.0E-6) do
                 begin
                        VmpuPrime := Vmpu;
                        Vmpu      := ln(1+k1) - ln(1 - VmpuPrime*ln(Io));
                        Vmpu      := Vmpu * (1-yr)/ln(k1);
                 end;

                Impu := Vmpu * XY(k1,Vmpu-1) * ln(k1);

                Vmp := Vmpu * Voc;
          Vmpu := Vmp/VBase;

           end
          else
           begin
                Ipu  := 0;
                Vpu  := 0;
                Vmpu := 0;
                Impu := 0;
           end;


{ Now, calculate the power values and write all of this to the
output file.}

          Ppu     := Ipu  * Vpu;
          Pmpu    := Impu * Vmpu;

          EN  := EN  + Ppu  * DayInt;
          ENmax := ENmax + Pmpu * DayInt;

          if (Is[I] = 0.0) then
                Is[I] := 0.0;
          writeln(A,Is[I]:5:3,'    ',Ppu:5:3);
    end;
    if ENmax > 0.0 then
          Eff  := EN/ENmax
     else
          Eff  := 0.0;
    ENmax := ENmax/EBase;
    EN  := EN/EBase;

    Loss   := ENmax - EN;
```

```
            DotLoc := pos('.',FileName) - 6;

            if copy(FileName,DotLoc,1) = 'A' then
                  Trint.Value := 100
             else
                  val(copy(FileName,DotLoc,2),TrInt.Value,Code);

            if copy(FileName,DotLoc+2,1) = 'A' then
                  PerCoInt.Value := 100
             else
                  val(copy(FileName,DotLoc+2,2),PerCoInt.Value,Code);

            if copy(FileName,DotLoc+4,1) = 'A' then
                  PerClInt.Value := 100
             else
                  val(copy(FileName,DotLoc+4,2),PerClInt.Value,Code);

            if RangeData then
                  write(G,TrInt.Value:3,PerCoInt.Value:5,PerClInt.Value:5,'   ');

            writeln(G,Rpu:5:3,'   ',EN:10:6,'   ',ENmax:10:6,'   ',Eff:10:6,'   ',Loss:8:4);

            Rpu := Rpu + Rinc;
      end;
      writeln(A,'Rpu = ',Rpu:5:3,'  Io = ',Io:8);
      close(A);
      close(G);
 end;


{    This procedure finds the maximum power points and optimum
       resistance values.
}
procedure MPPoint;

var
      Voc             :single;
      LogC                    :single;
      Is                      :single;
      X,XPrime                :single;
begin

      assign(G,'\solar\fill.out');
      rewrite(G);

      DataLabel := 'Maximum Power Points';
      writeln(G,DataLabel);


      Io := 1.0E-13;
```

```
while (Io < 1.0E-1) do
begin
        writeln(G,'');

        Voc   := 1.38049E-23 * Temp / 1.60219E-19* ln(1 + 1/Io);


        writeln(G,'Io = ',Io:8,'  Voc = ',Voc:6:4);
        writeln(G,'   Is    Vmpu     Impu    Rmpu     Pmpu');

        for I := 20 downto 1 do
        begin
                Is          := I/20;
                LogC        := 1+Is/Io;
                X           := 1;
                XPrime := 0;
                if (LogC <> 0) then
                begin
                        while (abs(X - XPrime) > 1.0E-5) do
                         begin
                                XPrime := X;
                                X            := ln(LogC/(1+XPrime));
                         end;
                        Vmpu     := 1.38049E-23 * Temp * X / 1.60219E-19;
                        Vmpu  := Vmpu / Voc;
                        Impu  := Is - Io * (exp(X) - 1);
                        Rpu   := Vmpu/Impu;

{    Calculate the maximum power point corresponding to each value
     of Is.}



                end
                else
                begin
                        Vmpu := 0;
                        Impu := 0;
                        Rpu  := 0;
                end;


{ Now, calculate the power values and write all of this to the
output file.}

                Pmpu := Impu * Vmpu;

                writeln(G,Is:8:4,Vmpu:8:4,Impu:8:4,Rpu:8:4,Pmpu:8:4);
        end;
```

```
            Io := Io*10;
        end;
        writeln(G,'');
        close(G);

 end;



procedure RangeInit;

begin

        FirstTime := true;
        TrInt.Value     := 0;
        PerCoInt.Value := 0;
        PerClInt.Value := 0;
        TrInt.Start     := 0;
        PerCoInt.Start := 0;
        PerClInt.Start := 0;
        TrInt.Stop      := 10;
        PerCoInt.Stop  := 10;
        PerClInt.Stop  := 10;
end;



{       This procedure generates files of insolation data according
                pplied values of transmittance, percent cover, clear
                ntage, and if a fixed system is used, the cloud size and speed.
}
procedure DayGen(NameFlag: char);
type
        CloudParam = record
                                a1r, a2r, a3r :real;
                        end;

var
        DayCount,CloudCount,ClearCount,IntCount :word;
        AllOK                                   :boolean;
        LengthOfDay                             :single;
        a1, a2, a3,Fdist,Area,AreaLast          :single;
        Small, Medium, Large                    :CloudParam;
begin

        FirstTime := TRUE;
        FileName := GetFileName(NameFlag,FirstTime);

        Randomize;
```

```
if (Condition <> 'A') then
 begin
        rewrite(L);
        writeln(L,Condition,' ',MaxVal);
        writeln;
 end;

write('Length of day (hours) :  ');
if FileRead then
 begin
        read(W,LengthOfDay);
        writeln(LengthOfDay:3:1);
 end
else
        readln(LengthOfDay);

if not Tracking then
 begin

        case CloudSize of

          'S': begin
                         a1 := 0.88;
                         a2 := 0.442;
                         a3 := 0.18;
                    end;
          'M': with Medium do
                    begin
                         a1 := 0.346;
                         a2 := 0.104;
                         a3 := 0.059;
                    end;
          'L': with Large do
                    begin
                         a1 := 0.147;
                         a2 := 0.033;
                         a3 := 0.022;
                    end;
          end;
 end;

while (TrInt.Value <= TrInt.Stop) do
begin

  PerCoInt.Value := PerCoInt.Start;

  while (PerCoInt.Value <= PerCoInt.Stop) do
  begin

    PerClInt.Value := PerClInt.Start;

    while (PerClInt.Value <= PerClInt.Stop) do
    begin
```

```
FileName := GetFileName(NameFlag,FirstTime);

LastDataFile := FileName;
rewrite(H);
writeln(H,LastDataFile);
writeln(H,LastPPUFile);
close(H);
if (Condition <> 'A') then
        writeln(L,FileName);
assign(F,FileName);
rewrite(F);
AllOk := not ErrOut(FileName);

Transmitt := TrInt.Value/100.0;
PerCover  := PerCoInt.Value/100.0;

PerClear  := PerClInt.Value/100.0;



writeln(F,'Solar Data, Trans. = ',TrInt.Value:2,  ', % Cover = ',PerCoInt.Value:2,',
        % Clear = ',PerClInt.Value);
DayCount := round(LengthOfDay*3600/DayInterval);
NumPts   := DayCount;
writeln(F,NumPts,' ',DayInterval);
IntCount := 90;
E        := 0.0;

if not Tracking then
        while DayCount > 0 do
         begin
                Fdist := random;
                AreaLast := 0.0;
                Area     := 0.5;
                Fdist    := random;
                while abs(Area - AreaLast) > 0.001 do
                 begin
                        AreaLast := Area;
                        Area := (Fdist + a3)/(a1 - a2*ln(AreaLast));
                    end;
                CloudCount := round(sqrt(3433609*Area)/Velocity);
                CloudCount := round(CloudCount/DayInterval);
                if CloudCount = 0 then
                        CloudCount := 1;
                if PerCover > 0.0 then
                       ClearCount := trunc(CloudCount*(1/PerCover - 1))+1
                 else
                        ClearCount := 50;
                while ClearCount > 0 do
                 begin
                       Z := DayCount*Pi/NumPts;
```

```
                                        Insr:= sin(Z)*PerClear;
                                        if CloudCount > 0 then
                                         begin
                                              Insr := Insr*Transmitt;
                                              dec(CloudCount);
                                         end
                                         else
                                              dec(ClearCount);
                                        if DayCount > 0 then
                                         begin
                                              writeln(F,Insr:6:3);
                                              dec(DayCount);
                                         end;
                                   end;
                              end
                    else
                    begin
                                   CloudCount := round(PerCover*DayCount);
                                   while DayCount > 0 do
                                    begin
                                        Insr := PerClear;
                                        if CloudCount > 0 then
                                         begin
                                             dec(CloudCount);
                                             Insr := Transmitt*Insr;
                                         end;
                                        dec(DayCount);
                                        writeln(F,Insr:6:3);
                                   end;
                    end;
                   close(F);

                   with PerClInt do
                         Value := Value + Increment;
          end;

          with PerCoInt do
                    Value := Value + Increment;
           end;
          with TrInt do
                   Value := Value + Increment;
          end;
          if (Condition <> 'A') then
                   close(L);
end;



{    This procedure displays a menu prompting for which data
         file(s) to plot.  The GetFileName procedure is then called,
         and the resulting file name is passed to the PlotData procedure.
```

```
}
procedure PlotRoutine;

var
        Choice              :char;
        Quit                :boolean;

begin
        ClrScr;
        writeln('1.     Plot most recently generated data file');
        writeln('2.     Select by file name');
        writeln('3.     Plot most recently generated data range');
        writeln('4.     Select by parameters');
        writeln;

        FirstTime := TRUE;
        FirstPlot := TRUE;
        Quit := FALSE;
        Choice := '0';
        while ((Choice < '1') or (Choice > '3') and (Choice <> 'X')) do
         begin
                readln(Choice);
                Choice := UpCase(Choice);
                case Choice of

                  '1': PlotData(LastDataFile,Quit);
                  '2': PlotData('',Quit);
                  '3': begin
                                reset(L);
                                readln(L,Condition,MaxVal);
                                while not Eof(L) do
                                 begin
                                        readln(L,FileName);
                                        PlotData(FileName,Quit);
                                 end;
                                close(L);
                        end;

                  '4': begin
                                RangeInit;
                                writeln;
                                write('(T)racking or (F)ixed  ');
                                readln(Choice);
                                if (Choice='T') or (Choice='t') then
                                        Tracking := TRUE;
                                while (TrInt.Value <= TrInt.Stop) do
                                begin

                                        PerCoInt.Value := PerCoInt.Start;

                                        while (PerCoInt.Value <= PerCoInt.Stop) do
                                         begin
```

```
                                        PerClInt.Value := PerClInt.Start;

                                        while (PerClInt.Value <= PerClInt.Stop) do
                                        begin
                                         if not Quit then
                                PlotData(GetFileName('X',FirstTime),Quit);
                                         with PerClInt do
                                                Value := Value + Increment;
                                        end;

                                        with PerCoInt do
                                                Value := Value + Increment;
                                end;

                                with TrInt do
                                        Value := Value + Increment;
                        end;

                end;
        end;



    end;
    CloseGraph;
{   TextMode(LastMode);}

end;



{   This procedure is called if a circuit is to be analyzed.  It
        prompts for a (range) of data file(s) using the GetFileName
        procedure, and then calls MaxPU with the proper parameters.
}
procedure Calculate;

var
        Choice,ChR          :char;
        Prompt              :boolean;
        FileName            :string;

begin

        ClrScr;
        writeln('1.     Use most recently generated data file');
        writeln('2.     Select by file name');
        writeln('3.     Use most recent data range');
        writeln('4.     Select by parameters');
        writeln;
```

```
FirstTime  := TRUE;
Prompt     := TRUE;
UserName   := FALSE;
SaveValue  := TRUE;
OutFile    := '';

Choice := '0';
while ((Choice < '1') or (Choice > '4') and (Choice <> 'X'))
do
 begin
        if FileRead then
                read(W,Choice)
         else
                readln(Choice);
        Choice := UpCase(Choice);
        if ((Choice > '0') and (Choice < '5')) then
         begin
                writeln;
                write('Vary R values ?  ');
                if FileRead then
                 begin
                        read(W,ChR);
                        writeln(Chr);
                 end

                else
                        readln(ChR);
                if ((ChR = 'Y') or (ChR = 'y')) then
                 begin
                        write('Initial R =        ');
                        readln(Ri);
                        write('Final R =          ');
                        readln(Rf);
                        write('Increment =        ');
                        readln(RInc);
                        write('Io =               ');
                        readln(Io);
                        RangeData := FALSE;
                 end
                 else
                  begin
                        write('R =  ');
                        if FileRead then
                         begin
                                read(W,Ri);
                                writeln(Ri:5:3);
                         end

                        else
                                readln(Ri);
                        write('Io = ');
                        if FileRead then
                         begin
```

```
                read(W,Io);
                writeln(Io:8);
            end
            else
                readln(Io);
            Rf := Ri;
            Rinc := 0.0;
            RangeData := TRUE;
        end;

end;

case Choice of

  '1': MaxPU(LastDataFile,Prompt);
  '2': MaxPU('',Prompt);
  '3': begin
            SaveValue := FALSE;
            reset(L);
            readln(L,Condition,MaxVal);
            while not Eof(L) do
             begin
                readln(L,FileName);
                MaxPU(FileName,Prompt);
             end;
            close(L);
        end;
  '4': begin
             RangeInit;
             FileName := GetFileName('X',TRUE);
             while (TrInt.Value <= TrInt.Stop) do
             begin

                  PerCoInt.Value := PerCoInt.Start;

                  while (PerCoInt.Value <=
                                         PerCoInt.Stop) do
                  begin

                       PerClInt.Value := PerClInt.Start;

                       while (PerClInt.Value <=
                                         PerClInt.Stop) do
                       begin
                          MaxPU(GetFileName('X',FALSE),Prompt);
                              with PerClInt do
                                  Value := Value +
                                               Increment;
                       end;

                       with PerCoInt do
                              Value := Value + Increment;
                  end;
```

```
                                        with TrInt do
                                                Value := Value + Increment;
                                end;

                        end;
                end;
        end;
end;




{       This procedure changes the time increment used when
        generating insolation files.
}

procedure ChangeSetup;

var
        Choice          :char;

begin
        writeln;
        write('Time increment for insolation calculations (seconds): ');
        readln(DayInterval);
        writeln;
end;




{       This procedure converts a file generated by MaxPU into one
        which is more presentable.
}
procedure MakePRT;

const
        BeginLabel      = '      Circuit Output, ';
        LineLabel       =
'----------------------------------------------------------------';
        TRLab           = '%Tr = ';
        COLab           = '%Cov = ';
        CLLab           = '%Clr = ';
        TRNext          = '  %Tr ';
        CONext          = '  %Cov';
        CLNext          = '  %Clr';

var
        NextLabel               : string;
        AllOK                   : boolean;
```

```
        Choice                      : char;
        Dummy                       : string[15];
        IntTR,IntCO,IntCL,Val,i     : integer;
        Value                       : single;
        FirstData                   : Boolean;
        DummyReal                   : single;
        PRTFile                     : string;

procedure ListIt;

begin
        write(A,'   ');

        if Condition = 'T' then
         begin
                write(A,IntTR:3);
                Val := IntTR;
         end;

        if Condition = 'O' then
         begin
                write(A,IntCO:3);
                Val := IntCO;
         end;

        if Condition = 'L' then
         begin
                write(A,IntCL:3);
                Val := IntCL;
         end;


        write(A,'  |  ');

        writeln(A,EN:6:4,'      ',ENmax:6:4,'      ',Eff:6:4,'      ',Loss:6:4);

end; {ListIt}


begin

        FirstData := TRUE;

        Done := FALSE;
        if FileName = '' then
         begin
                writeln;
                write('.PPU File: ');
                readln(FileName);
         end;
        if ((FileName = 'L') or (FileName = 'l')) then
         begin
                reset(H);
```

```
        readln(H,FileName);
        readln(H,FileName);
        close(H);
 end;
if (Pos('.',FileName) = 0) then
        FileName := FileName + '.ppu';

assign(F,FileName);
reset(F);
PRTFile := FileName;
delete(PRTFile,pos('.',PRTFile)+1,3);
PRTFile := PRTFile + 'PRT';
assign(A,PRTFile);
rewrite(A);


if not ErrOut(FileName) then
 begin
         while (not Eof(F)) do
          begin
                NextLabel := '        EN        ENmax        Eff        Loss';

                readln(F,DataName);
                readln(F,DataName);
                readln(F,Condition,MaxVal);
                case Condition of
                 'T': DataName := 'varying % Cloud Tran.';
                 'L': DataName := 'varying % Clear Sky';
                 'O': DataName := 'varying % Cloud Cover';
                 end;
                DataName := BeginLabel + DataName;
                writeln(A,DataName);
                writeln(A,'');
                readln(F,Dummy);
                write(A,Dummy,', R = ');
                for i := 1 to 3 do
                        readln(F,Param);
                i := 0;

                AllOK  := errout(FileName);

                readln(F,IntTR,IntCO,IntCL,R,EN,ENmax,Eff,Loss);
                write(A,R:5:3,', ');

                if Condition <> 'T' then
                 begin
                        write(A,TRLab,IntTR,', ');
                        inc(i);
                 end
                 else
                        NextLabel := TRNext + NextLabel;
```

```
                    if Condition <> 'O' then
                     begin
                            write(A,COLab,IntCO);
                            inc(i);
                            if i = 1 then
                                    write(A,', ');
                     end
                     else
                            NextLabel := CONext + NextLabel;

                    if Condition <> 'L' then
                            write(A,CLLab,IntCL)
                     else
                            NextLabel := CLNext + NextLabel;

                    writeln(A,'');
                    writeln(A,'');
                    writeln(A,NextLabel);
                    writeln(A,LineLabel);

                    ListIt;

                    while Val < MaxVal do
                     begin
                            readln(F,IntTR,IntCO,IntCL,R,EN,ENmax,Eff,Loss);
                            ListIt;
                     end;

                    for i := 1 to 3 do
                            writeln(A,'');

             end;
            close(F);
            close(A);
            Done := False;
        end;
end;



{ Initialization Code}
{ This section of code merely initialized the proper variables.}

begin

        Temp := 300; {K}
         DayInterval := 300;  {Seconds}
        Param := FSearch('LASTFILE.FIL','c:\THESIS');
        if (Param = '') then
         begin
                assign(H,'\thesis\lastfile.fil');
```

```
{ Initialization Code}
{ This section of code merely initialized the proper variables.}

begin

        Temp := 300; {K}
         DayInterval := 300;  {Seconds}
        Param := FSearch('LASTFILE.FIL','c:\THESIS');
        if (Param = '') then
         begin
                assign(H,'\thesis\lastfile.fil');
                rewrite(H);
         end
        else
         begin
                Assign(H,FExpand(Param));
                reset(H);
                readln(H,LastDataFile);
                readln(H,LastPPUFile);
        end;
        close(H);

{$IFDEF BigProg}


        Param := FSearch('LASTRNG.FIL','c:\THESIS');
        if (Param = '') then
                assign(L,'\thesis\lastrng.fil')
        else
                Assign(L,FExpand(Param));


        GetMem(PSR,sizeof(PSrec));
        FillChar(PSR^,sizeof(PSrec),0); {clear prnscr record to zero}
        with PSR^ do
        begin
                GPage := 0;             {use graphics page 0}
                LPTnum := 0;             {assume printer on LPT1}
                ScrnType := GraphDriver;  {use turbo's driver number}
                PStype := 5;            {use standard mode}
                LandScape := true;      {define the print mode (landscape/upright}
                mono := true;           {assume monochrome mode}

                PrnLArea.Xmin := 0;     {define the printer landscape defaults}
                PrnLArea.Ymin := 0;
                PrnLArea.Xmax := 479;
                PrnLArea.Ymax := 800;
                PrnUArea.Xmin := 0;      {define the printer upright defaults}
                PrnUArea.Ymin := 0;
                PrnUArea.Xmax := 959;
                PrnUArea.Ymax := 431;
                PrnArea := PrnUArea;     {start with upright print default}
                ScrnArea.Xmin := 0;     {define the screen defaults}
```

```
          ScrnArea.Ymin := 0;
          ScrnArea.Xmax := 639;   {VGA graphics assumed}
          ScrnArea.Ymax := 479;
          initprn(PSR^);          {now go initialize it}
     end;

     Code := RegisterBGIDriver(@EGAVGA);
     Code := RegisterBGIFont(@Small);
{$ENDIF}
end.
```

APPENDIX C

DERIVATION OF EQUATION FOR MAXIMUM POWER RESISTANCE

Repeating equation (11) from page 22:

$$I_L = I_S - I_O [\exp(eV/kT) - 1], \quad (11)$$

we find the power by multiplying both sides by the voltage V, which results in equation (19).

$$P = V I_L = V \{ I_S - I_O [\exp(eV/kT) - 1] \} \quad (19)$$

Taking the derivative of both sides with respect to V, and setting dP/dV to zero and grouping terms gives equation (20).

$$0 = I_S + I_O - I_O [(1 + eV/kT) \exp(eV/kT)] \quad (20)$$

After putting the current terms on one side of the equation, the voltage terms on the other, we get equation (21).

$$1 + I_S/I_O = (1 + eV/kT) \exp(eV/kT) \quad (21)$$

For each value of I_S and I_O, the terms on the left-hand side of equation (21) evaluate to a constant, and V can then be solved for iteratively. The value for V is then substituted into equation (11) to find I_L. Once I_L is found, $R_{PU,MP}$ is then given by

$$Rpu,mp = (V/I_L) / R_{BASE} \tag{22}$$

APPENDIX D

THE OPERATION OF LOADS POWERED BY SINGLE SOURCES OR BY A COMMON SOURCE OF SOLAR CELLS

THE OPERATION OF LOADS POWERED BY SINGLE SOURCES OR BY A COMMON SOURCE OF SOLAR CELLS

J. Appelbaum, Senior Member, IEEE
Faculty of Engineering, Tel-Aviv University
Tel-Aviv 69978, Israel.

Abstract - In many photovoltaic systems, the solar cell array is designed to power a specific single load. Several loads of the same or different types may be powered either by separate solar cell sources for each one, or alternatively, by a common solar cell source for all the loads; and for each one of these two possibilities, the system may include a maximum power point-tracker (MPPT). The purpose of this paper is to introduce a procedure for comparing these two possible setups. Loads that are connected to a common source interact with each other so that one load may improve its operation at the expense of another load. The criterion for comparing the performance of the loads is the "energy utilization efficiency". Two types of loads were considered as examples: (a) two different ohmic loads; and (b) two different water electrolyzer loads. The study shows: the performance of one load affects the performance of another load in a common source system; in systems not including MPPTs, the total performace of all the loads in a common source system is improved as compared to the performance of all the loads when they are powered separately by individual sources; in systems including MPPTs, the total performance of all the loads is the same when powered either by a common source or by single sources; and MPPTs improve the performance of loads, the amount depending on the mismatch of the loads to the solar cells.

INTRODUCTION

In many stand-alone photovoltaic systems, the solar cell (SC) array is designed to power a specific single load, such as a warning light, a water electrolyzer, or a water pump. Several loads of the same or different types that are in close proximity to each other may be powered either by a separate SC source for each one, or alternatively, by a common SC source for all the loads. Loads of either the same or different types may have different I-V characteristics, and since a SC source is a finite power one and has different I-V external characteristics from the conventional power source, it can be expected that the operation of the different loads will also be different when the loads are powered either by separate sources or by a common SC source. It is possible that the operation of one load may be improved at the expense of another load, or the total performance of all the loads powered by the common source can also be improved. In certain cases, such as at low solar insolation, it might be advantageous to disconnect a load from the system in order to improve the operation of another load.

The purpose of this study is to introduce a procedure for analyzing the interaction between different loads when they are connected to a common SC source, and also to compare the load performances when connected to separate sources. Two types of loads as examples are considered separately: (a) two different ohmic loads; (b) two different water electrolyzer loads. A comparison of the operation of the above two loads was also performed for systems that incorporate a maximum power point-tracker, for both the separate and common SC sources. The criterion for comparing the performance is the "energy utilization efficiency".

It is assumed that the operation of the loads takes place on a clear day with the variation of the solar insolation given by:

$$G = 100 \sin\omega \quad , \tag{1}$$

where G is the solar insolation in percentages of ONE SUN, and ω is the solar angle in degrees. The solar angle is related to the solar time by:

$$T = (\omega + 90)/15 \quad , \tag{2}$$

when at solar noon (T = 12:00) the solar angle is 90°. A "time utilization efficiency" η_T was defined in [1] by:

$$\eta_T = 100\ P/P_M \quad , \tag{3}$$

where P is the SC array output power, and P_M is its maximum output power; both are functions of the solar insolation G. The "energy utilization efficiency" of the solar cells in the system is defined here by:

$$\eta_E = \int_{T_1}^{12:00} P(T)dT \Big/ \int_{0}^{12:00} P_M(T)dT \quad , \tag{4}$$

where the numerator is the input energy to the load by the SC array during the operating period between some time T_1 and noon; and the denominator is the maximum available energy that the SC array can supply. The time T_1 corresponds to some threshold operating value of the load, e.g., current, voltage or power. Because of the assumed symmetry of the insolation around noon, the calculation is performed for half a day. The η_E indicates the degree of utilization of the solar cells in the system, or in other words, the degree of matching the load to the solar cells.

The SC array used in this study is represented by the following I-V approximate equation:

$$V = -IR_s + \frac{1}{A} \ln\left(\frac{I_{ph} - I + I_o}{I_o}\right) \quad , \tag{5}$$

where

I_{ph} is the photocurrent (amps.) proportional to the solar insolation;
I_o is the reverse saturation current;
R_s is the series resistance;
q is the electron charge;

A is the ideality factor;
k is the Bolzmann constant;
T is the absolute temperature; and
Λ = q/AkT.

A suitable array for the different single loads consists of 18 parallel strings, each string made up of 9 panels in series, and each panel having 36 series-connected cells. The parameter values of a single cell are I_o=0.45x10^{-3}A; Λ=13.68 1/V; R_s=0.05Ω; and I_{ph}=0.756 A (at the insolation of 1000 Wm^{-2}=100% insolation). The appropriate I-V equation of the array is:

$$V = -0.9\ I + \frac{1}{0.0422} \ln \left(\frac{I_{ph}-I+0.0081}{0.0081}\right) , \qquad (6)$$

where for 100% insolation, I_{ph}=13.615 A; V_{oc}=176 V; and P_M=1400 W. The common SC source is made up of two equal single sources (eq. (6)) connected in parallel. The I-V equation for this array is:

$$V = -0.45\ I + \frac{1}{0.0422} \ln \left(\frac{I_{ph}-I+0.0162}{0.0162}\right) , \qquad (7)$$

where for 100% insolation, I_{ph}=27.23A; V_{oc}=176V; and P_M=2800 W.

It is necessary to point out that for identical loads of the same type and size powered by identical SC arrays, and for loads of the same type but different in size that are powered by proportional sizes of SC arrays, the operation of the loads are the same when powered either by separate sources or by a common SC source. The operation of the loads are expected to be different for cases as: loads of different types powered by appropriate SC array sizes; loads of the same type but different in their size powered by identical SC arrays; loads of the same type powered by SC arrays of the same size but having variations either in their load or array parameters. In general, differences in the load and SC array characteristics may inevitably be introduced in the system design or during the operation leading to different operation of the loads when powered by single sources or by a common source of solar cells.

MAXIMUM POWER POINT-TRACKER (MPPT)

For maximum utilization of the solar cells, the matching of the load to the solar cell array is accomplished by incorporating into the system an electronic control device - a maximum power point-tracker. The MPPT may be viewed as a time-variable transformer (TVT) [2], in which the transformation ratio 'n' is changed electronically, corresponding to the variations of the load operating point due to the variations of the solar insolation. Fig. 1a represents a system consisting of an SC array; an MPPT with a ratio of 1:n; and a load. The array operates at its maximum power output, P_M=$V_M I_M$, where V_M and I_M are the voltage and current at maximum power, respectively. This power is delivered to the load by means of the MPPT (assuming 100% efficiency of the MPPT), i.e.,

$$P_M = V_M I_M = V_L I_L . \qquad (8)$$

Loads of different types may be represented (using Thevenin's theorem) by an equivalent voltage V_o in series with an equivalent resistance R. Using the TVT transformation (see Fig. 1b), one may write:

$$V_M = V_o' + I_M R' = V_o/n + I_M R/n^2 , \qquad (9)$$

for which the solution for n is:

$$n = \frac{V_o}{2V_M} + \left[\left(\frac{V_o}{2V_M}\right)^2 + \frac{I_M R}{V_M}\right]^{1/2} . \qquad (10)$$

The load voltage V_L and the load current I_L are related to V_M and I_M by:

$$V_L = nV_M \quad \text{and} \quad I_L = I_M/n , \qquad (11)$$

and

$$P_M = V_o I_L + I_L^2 R . \qquad (12)$$

In the following section, each type of load is analyzed separately for the following cases, and conclusions are drawn for:

1. A system without an MPPT where a single load is powered by a single source;
2. A system without an MPPT where two loads are powered by a common source;
3. A system with an MPPT where a single load is powered by a single source; and
4. A system with an MPPT where two loads are powered by a common source.

OHMIC LOAD

Two ohmic loads of different size but designed according to a common criteria are connected thusly; the first time separately to a single SC array (eq. (6)); and the second time in parallel to a double-sized common SC array, (eq. (7)), as shown in Figs. 2a and 2b, respectively. These two cases are analyzed again when connected into systems containing MPPTs (Figs. 7a

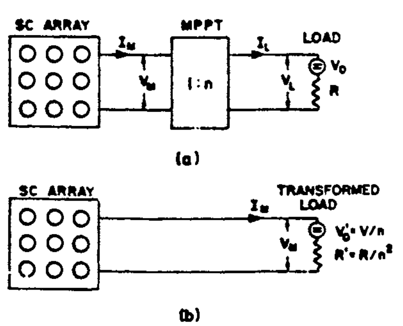

Fig. 1. Photovoltaic system:
(a) system consisting of a solar cell array, an MPPT, and a load;
(b) the equivalent

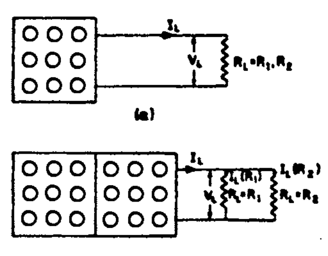

Fig. 2 Ohmic loads:
(a) single load and single source;
(b) combined load and common source.

& 7b). The load resistances of $R_1 = R_1 = 8.16\Omega$ and $R_2 = R_2 = 15.18\Omega$ were determined for equal input power of 1300 watts at one SUN (see Fig.3, points c and f). The threshold operating points of these loads are at a and d, again, corresponding to equal input power of 140 watts. Powers of less than 140 watts for each load are considered unuseful powers.

1. Operation of a System Without an MPPT for a Single Ohmic Load and a Single Source

The I-V characteristic of an ohmic load is given by:

$$V_L = I_L R_L \quad . \tag{13}$$

The I-V characteristics of the SC array for different levels of insolation, in percentages of ONE SUN, computed by eq. (6), are shown in Fig. 3. The figure also includes the maximum power line P_M, and the two load lines defined by eq. (13). The system operation is determined by the intersection point of the load line with the I-V characteristics of the SC source for various insolation levels, i.e., points a, b, and c. The load current is obtained by equating eqs. (6) and (13), and solving I (using a computer library program), i.e.,

$$-0.9I_L + \frac{1}{0.0422}\ln\left(\frac{I_{ph} - I_L + 0.0081}{0.0081}\right) = R_1 I_L \text{ and } R_2 I_L . \tag{14}$$

The corresponding load voltage V_L is then computed from eq. (13), and the load input power is $P_L = V_L I_L$. Fig. 4 describes the performance of these two loads as

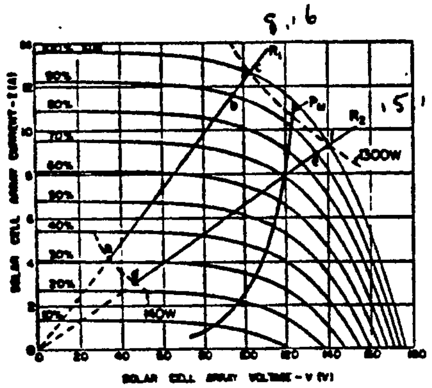

Fig. 3. I-V characteristics of the single SC source and the separate ohmic loads.

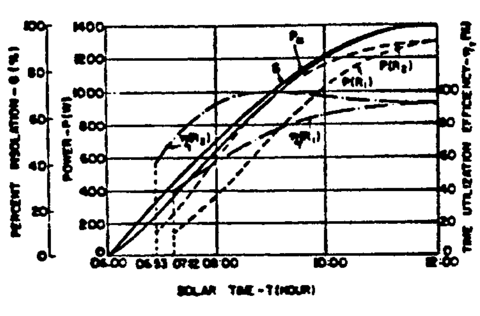

Fig. 4. Time utilization efficiences of the separate ohmic loads without an MPPT.

a function of the solar time T (eq. (2)), from sunrise (T=06:00) to noon (T=12:00). The Fig. 4 includes: the solar insolation G; the array maximum power P_M; the two load input powers $P(R_1)$ and $P(R_2)$; and the load time utilization efficiencies $\eta_T(R_1)$ and $\eta_T(R_2)$ as defined by eq. (3). The energy utilization efficiencies are calculated by eq. (4) with the help of a computer library program based on the theory of SPLINES [3], i.e.,

$$\eta_E(R_1) = \int_{07:12}^{12:00} P(R_1)dT \Big/ \int_{0}^{12:00} P_M dT = 78.44\%$$

$$\eta_E(R_2) = \int_{06:53}^{12:00} P(R_2)dT \Big/ \int_{0}^{12:00} P_M dT = 92.76\% \tag{15}$$

The average of $\eta_E(R_1)$ and $\eta_E(R_2)$ is:

$$\eta_E(R_1 + R_2) = 85.6\% \tag{16}$$

The utilization of the cells by the load R_2 is higher than by R_1.

2. Operation of a System Without an MPPT for Two Ohmic Loads and a Common Source

The two ohmic loads are now connected to a common source as shown in Fig. 2b. The SC array current is obtained from the solution of eq. (7) and from the parallaly connected load, given by:

$$-0.45I_L + \frac{1}{0.0422}\ln\left(\frac{I_{ph} - I_L + 0.0162}{0.0162}\right) = (R_1 || R_2) I_L \quad , \tag{17}$$

for which the voltage of the loads is $V_L = (R_1||R_2)I_L$, and the load currents are $I_L(R_1) = V_L/R_1$, and $I_L(R_2) = V_L/R_2$. The I-V characteristics of the SC array, of the combined load $R_L = R_1||R_2$, and of each load in the combined system are shown in Fig. 5. The solution is again for $P(R_1)$ and $P(R_2) \geq 140W$. Since the combined load shares a common voltage, the threshold power of 140W is first reached by load R_1 at point a (06:55 solar time). With the increase of the insolation, load R_2 starts to operate at point d (07:17 solar time). Fig. 6 describes the load powers $P(R_1)$ and $P(R_2)$; the combined load power $P(R_1||R_2)$; and the utilization efficiencies η_T of the loads as a function of the solar time. As is evident from Figs. 4 and 6, the connection of the two loads to a common source results in different performances of the loads,

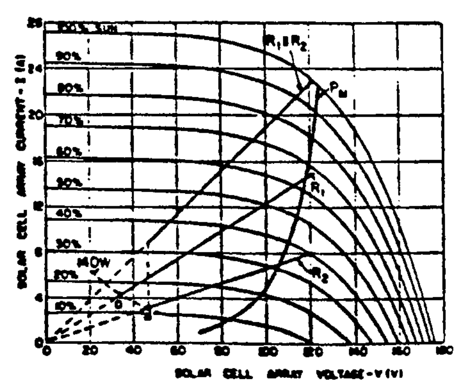

Fig. 5. I-V characteristics of the common SC source and the combined ohmic loads.

compared to the performance of the same loads when powered from single sources. The energy utilization efficiencies are:

$$\eta_E(R_1)=58.60\%;\quad \eta_E(R_2)=31.12\%;$$
and $$\eta_E(R_1||R_2)=89.72\% \quad . \tag{18}$$

By comparing the two possible connections of the loads, the first time to separate sources and the second time to a common source, the conclusions (eqs. (15), (16) and (18)) are:

(a) the operation of the load $R_L=8.16\Omega$ is improved by $58.60 - 78.44/2 = 19.38\%$;
(b) the operation of the load $R_L=15.18\Omega$ deteriorates by $31.12 - 92.76/2 = 15.26\%$; and
(c) the operation of the combined load $R_L=R_1||R_2$ is improved by $89.72 - 85.60 = 4.12\%$.

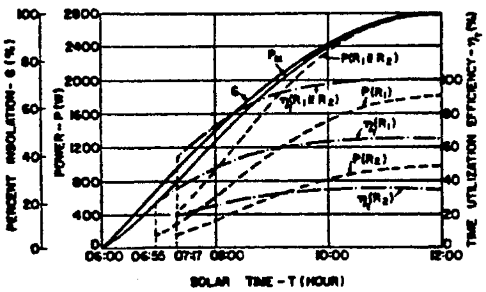

Fig. 6. Time utilization efficiency of the combined ohmic loads without an MPPT.

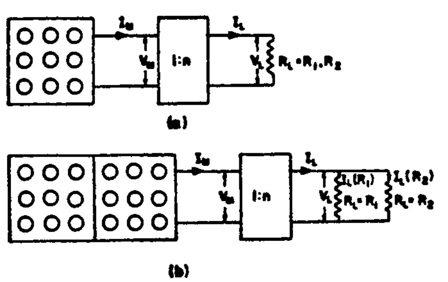

Fig. 7. A photovoltaic system with an MPPT:
(a) single sources and separate ohmic loads;
(b) common source and combined ohmic loads.

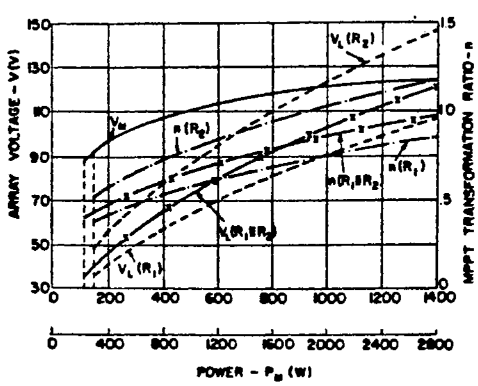

Fig. 8. Variation of load voltages, and transformation ratios of single and common sources for ohmic loads.

The efficiencies in the above calculations are divided by the factor 2 because the comparison is made for single loads once connected to a single source and second time connected to a double-sized source.

3. Operation of a System With an MPPT for a Single Ohmic Load and a Single Source

Referring to Fig. 1a and eq. (10), $V_o=0$ for an ohmic load; the transformation ratio of the MPPT thus becomes;

$$n = (I_M R_L / V_M)^{1/2} \quad , \tag{19}$$

and the load voltage, current, and power are: $V_L=nV_M$, $I_L=I_M/n$, and $P=P_M=I_L^2R_L$, respectively. The two loads, $R_L=R_1$ and $R_L=R_2$, are connected separately to single sources (eq. (6)), as shown in Fig. 7a. Again, the operation of the loads is constrained by $P(R_1)$ and $P(R_2)\geq 140W$. Since the MPPT (assuming 100% efficiency of the MPPT) matches the load line to the maximum power line of the SC array, the time utilization efficiency of the solar cells is 100% for the period of operation. On the other hand, the energy utilization efficiency η_E is less than 100%, because the solar time for which the loads start to operate depends on the threshold power (140W); this time for both loads is T=06:34. The energy utilization efficiencies, in this case, are the same for both loads:

$$\eta_E(R_1) = 99.25\% \ , \quad \eta_E(R_2) = 99.25\% \ , \tag{20}$$

and the average is:

$$\eta_E(R_1+R_2) = 99.25\% \ . \tag{21}$$

Shown in Fig. 8 as a function of the SC array maximum power P_M are: the variation of the transformation ratio $n=V_L/V_M$ ($n(R_1)$, $n(R_2)$); the load voltage V_L of each load, ($V_L(R_1)$, $V_L(R_2)$); and the voltage at maximum power P_M (single source - 1400 W). This figure shows that the variation of the load voltage V_L is larger than the variation of the array voltage V_M. By comparing the performance of the two systems, one with an MPPT and the other without it, both for single source operation, the conclusion (eqs. (15) and (20)) is in favor of using an MPPT since:

(a) the operation of load $R_L=8.16\Omega$ is improved by 20.81%; and
(b) the operation of load $R_L=15.18\Omega$ is improved by 6.49%.

4. Operation of a System With an MPPT for Two Ohmic Loads and a Common Source

In our last example, the two ohmic loads are connected to the common source (eq. (7)) via the MPPT, as shown in Fig. 7b. The transformation ratio 'n' is given in eq. (19), where the load resistance R_L is the resistance of the parallel connection of R_1 and R_2. The equivalent load voltage is $V_L=nV_M$, and the current is $I_L=I_M/n$. The powers are:

$$P = P_M = P_M(R_1) + P_M(R_2) = I_L^2 R_L(R_1||R_2);$$
$$P_M(R_1) = I_L(R_1)V_L = I_L^2(R_1)R_L(R_1) \quad ; \text{ and} \tag{22}$$
$$P_M(R_2) = I_L(R_2)V_L = I_L^2(R_2)R_L(R_2),$$

where $I_L(R_1)=V_L/R_L(R_1)$, and $I_L(R_2)=V_L/R_L(R_2)$. Again, the solution is for $P(R_1)$ and $P(R_2)\geq 140W$; and for 100% efficiency of the MPPT. Because of the load matching, the time utilization efficiency of the combined load for the period of operation is 100%; for $R_1=8.16\Omega$ it is 65.04%; and for $R_2=15.18\Omega$ it is 34.96%, as shown in

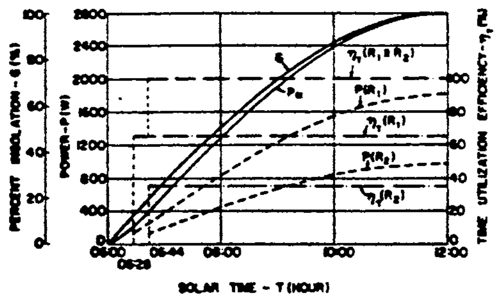

Fig. 9. Time utilization efficiency of the combined ohmic loads with an MPPT.

Fig. 9. The threshold power of 140 Watts is obtained at the solar time of 06:28 for R_1 and at 06:44 for R_2. The energy utilization efficiences are:

$$\eta_E(R_1) = 64.75\% \; ; \; \eta_E(R_2) = 34.50\% \; ;$$
and
$$\eta_E(R_1||R_2) = 99.25\% \quad . \tag{23}$$

Shown in Fig. 8 as a function of the SC array maximum power P_M (common source—2800W) are: the variation of the transformation ratio $n(R_1||R_2)$; the load voltage $V_L(R_1||R_2)$; and the voltage at maximum power V_M.

The following conclusions are drawn:

(a) By comparing eqs. (18) and (23), the performances of two systems with a common source, one with an MPPT and the other without, we see that there is an improvement of 9.53% in the system performance that included an MPPT.

(b) By comparing eqs. (21) and (23), the performance of two systems with single sources and a common source, both systems including an MPPT this time, we see that the improvement is the same for both systems, due to both having an MPPT.

(c) By comparing eqs. (20) and (23), where again there are single sources and a common source, and where the MPPT is included both times, but where the comparison is this time between individual loads, we see that there is an improvement of 15.13% in the performance of the load R_L=8.16Ω, at the expense of load R_L=15.18Ω, in the common source system.

WATER ELECTROLYZER LOADS

Two water electrolyzer loads, WE1 and WE2, are separately connected, the first time to two single SC arrays, (eq. (6)), and the second time to a double-sized common array, (eq. (7)), as shown in Figs. 10a and 10b.

1. Operation of a System Without an MPPT for a Single Water Electrolyzer and a Single Source

The I-V characteristics of a water electrolyzer may be represented by the equation:

$$V = V_o + IR_p \quad , \tag{24}$$

where V_o is the open circuit voltage, and R_p is the internal resistance. Fig. 11 shows the I-V characteristics of the single SC array (eq. (6)), and the straight lines of the two electrolyzers given here by:

$$(WE1) \; , \; V = 70 + 4.82 \; I \quad ,$$
$$(WE2) \; , \; V = 90 + 3.45 \; I \quad , \tag{25}$$

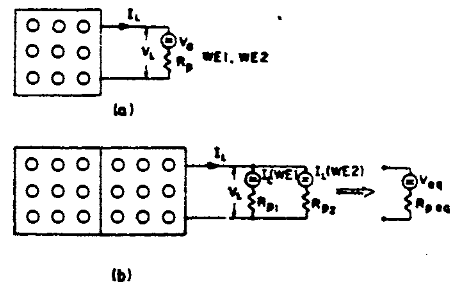

Fig. 10. Water electrolyzer load:
(a) single load and single source;
(b) combined load and common source.

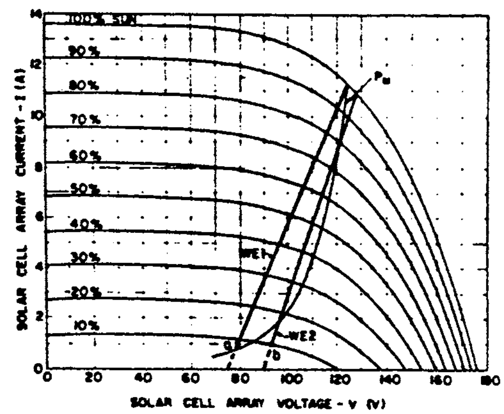

Fig. 11. I-V characteristics of the single SC source and the separate electrolyzer loads.

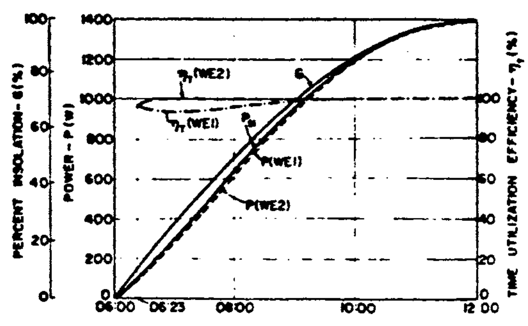

Fig. 12. Time utilization efficiencies of the separate electrolyzer loads without an MPPT.

where

$$V_{o1}=70 \text{ V} \; , \; V_{o2}=90\text{V} \; , \; R_{p1}=4.82Ω, \text{ and } R_{p2}=3.45Ω \quad . \tag{26}$$

It is assumed that the electrolyzers start to operate at insolation levels when sufficient current density has been obtained, i.e., at threshold current of one ampere (see points a and b). Good matching of an electrolyzer to the SC array can be obtained by carefully selecting the number of cells in the electrolyzer, as shown in Fig. 11 by the close proximity of the load lines to the maximum power line

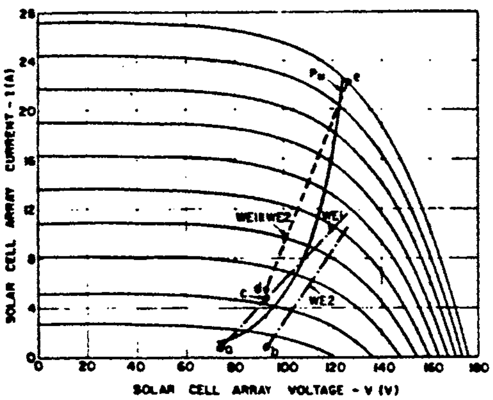

Fig. 13. I-V characteristics of the common SC source and the combined electrolyzer loads.

P_M. The current is obtained by equating eqs. (6) and (25), and solving I, i.e.,

$$-0.9I_L+\frac{1}{0.0422}\ln\left(\frac{I_{ph}-I_L+0.0081}{0.0081}\right)=70+4.82I_L \text{ and } 90+3.45I_L . \quad (27)$$

The load voltage is given by eq. (25), and the load power is $P_L=V_LI_L$. Fig. 12 describes the performance of these two loads as functions of the solar time T. The figure includes: the solar insolation G; array power P_M; the two load input powers P(WE1) and P(WE2); and the load time utilization efficiences η_T(WE1) and η_T(WE2). Because of good matching between the loads and the SC array, the energy utilization efficiences are high (the calculations follow the procedure as described for the ohmic load):

$$\eta_E(WE1)=98.56\%, \text{ and } \eta_E(WE2)=99.48\% . \quad (28)$$

The energy utilization efficiency of both electrolyzers is:

$$\eta_E(WE1 + WE2) = 99.02\% . \quad (29)$$

2. Operation of a System Without an MPPT for Two Water Electrolyzers and a Common Source

The above-mentioned two electrolyzers are connected in parallel to a common source, as shown in Fig. 10b. The equivalent of the parallel connection of these two loads is:

$$V = V_{oeq} + IR_{peg} . \quad (30)$$

where

$$V_{oeq}=V_{o1}\frac{R_{p2}}{R_{p1}+R_{p2}}+V_{o2}\frac{R_{p1}}{R_{p1}+R_{p2}}, \text{ and } R_{peq}=\frac{R_{p1}R_{p2}}{R_{p1}+R_{p2}} . \quad (31)$$

The array current is obtained from the solution of eqs. (7), (26), (30) and (31):

$$-0.45I_L+\frac{1}{0.0422}\ln\left(\frac{I_{ph}-I_L+0.0162}{0.0162}\right) = 81.66+2.01I_L . \quad (32)$$

The electrolyzer's voltage is $V_L=81.66+2.01I_L$, and the currents are:

$$I_L(WE1)=(V_L-V_{o1})/R_{P1} \text{ and } I_L(WE2)=(V_L-V_{o2})/R_{p2} . \quad (33)$$

The I-V characteristics of the SC array, the combined

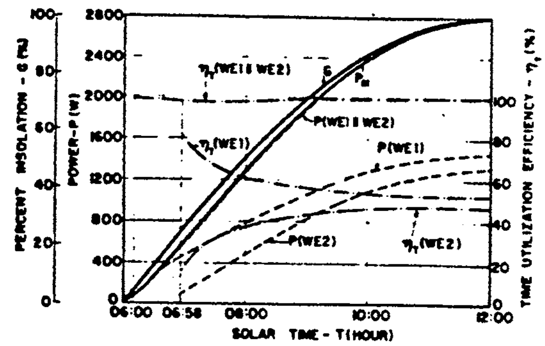

Fig. 14. Time utilization efficiency of the combined electrolyzer loads without an MPPT.

electrolyzer loads WE1‖WE2, and each load WE1 and WE2 in the combined system are shown in Fig. 13. For low insolation levels, for which I_L(WE2)<1.0 A, only WE1 operates (see line ac). For higher insolation levels, both the electrolyzers operate (see line de). Fig. 14 describes the electrolyzer's input powers P(WE1), P(WE2), and the combined power P(WE1‖WE2); and their utilization efficiencies η_T as a function of the solar time T. Again, the interaction between the loads and its effect on their operation in the combined system is evident (Figs. 12 and 14). The energy utilization efficiencies are:

$$\eta_E(WE1) = 56.45\% \ ; \ \eta_E(WE2) = 43.07\% ; \text{ and } \eta_E(WE1\|WE2) = 99.52\% . \quad (34)$$

By comparing the performance of the loads, the first time connected to separate sources, and the second time to a common source, the conclusions (eqs. (28), (29) and (34)) are:

(a) the operation of electrolyzer WE1 is improved by 7.17%;
(b) the operation of electrolyzer WE2 deteriorates by 6.67%; and
(c) the operation of combined electrolyzers WE1‖WE2 is improved by 0.5%.

3. Operation of a System With an MPPT for a Single Water Electrolyzer Load and a Single Source.

A single water electrolyzer system including an MPPT is represented in Fig. 1a, and the transformation ratio 'n' is given by eq. (10). The load voltage, current and power are $V_L=nV_M$, $I_L=I_M/n$, and $P=P_M=V_LI_L$, respectively; or

$$P_M = I_LV_o + I_L^2R_p . \quad (35)$$

Since the MPPT matches the load-line to the maximum power line of the SC array, the time utilization efficiency of the solar cells is 100% for the period of operation. On the other hand, the energy utilization efficiency η_E for WE1 and WE2 is less than 100%, because the solar time when the loads start to operate depends on the threshold current (1.0 A). The results are:

$$\eta_E(WE1) = 99.77\% ; \ \eta_E(WE2) = 99.63\% ; \text{ and } \eta_E(WE1 + WE2) = 99.70\% . \quad (36)$$

Shown in Fig. 15 as a function of the SC array maximum

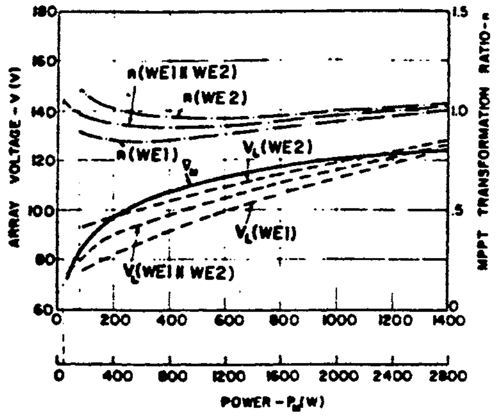

Fig. 15. Variation of the voltages, and transformation ratios of single and common sources for the electrolyzer loads.

power P_M are: the variation of the transformation ratio (n(WE1), n(WE2)); the load voltage V_L of each electrolyzer (V_L(WE1), V_L(WE2)); and the voltage at maximum power P_M (single source - 1400 W). Because the load lines are positioned close to the maximum power lines of the SC arrays, the variation of the load voltages with respect to V_M is small and hence, the variation of 'n' is also small. By comparing the performance of the two systems, one with an MPPT and the other without it, for single source operation, the results (eqs. (28) and (30)) show only a small improvement in using an MPPT:

(a) the operation of WE1 is improved by 1.21%; and
(b) the operation of WE2 is improved by 0.15%.

4. Operation of a System With an MPPT for Two Electrolyzer Loads and a Common SOurce

The two water electrolyzers WE1 and WE2 were connected in parallel to a common SC array source, as shown in Fig. 10b, but this time via an MPPT. The transormation ratio 'n' according to eq. (10) is:

$$n = \frac{V_{oeq}}{2V_M} + \left[\left(\frac{V_{oeq}}{2V_M}\right)^2 + \frac{I_M R_{peq}}{V_M}\right]^{1/2} , \quad (37)$$

where V_{oeq} and R_{peq} are given in eqs. (26) and (31), i.e., V_{oeq}=81.66 V and R_{peq}=2.01Ω. The combined load voltage, current and power are: V_L=nV_M; I_L=I_M/n; and P=P_M=$V_L I_L$, respectively, or

$$P_M = I_L V_{oeq} + I_L^2 R_{peq} . \quad (38)$$

The electrolyzers' currents are given in eq. (33), and their powers are:

$$P(WE1) = V_L I_L(WE1) = I_L(WE1)\, V_{o1} + I_L^2(WE1) R_{p1} ,$$
$$P(WE2) = V_L I_L(WE2) = I_L(WE2)\, V_{o2} + I_L^2(WE2) R_{p2} . \quad (39)$$

Because of load matching, the time utilization efficiency of the combined electrolyzer load for the period of operation is 100%. The variations of P(WE1), P(WE2), η_T(WE1), and η_T(WE2) are similar to those in Fig. 14. The energy utilization efficiencies are:

and
$$\eta_E(WE1)=56.66\% \; ; \; \eta_E(WE2)=43.20\% ;$$
$$\eta_E(WE1\|WE2)=99.86 . \quad (40)$$

The following conclusions are drawn:

(a) By comparing eqs. (34) and (40), the performance of two systems with a common source, one with an MPPT and the other without, we see that there is only a small improvement of 0.34% in the system performance that included the MPPT;
(b) By comparing eqs. (36) and (40), the performance of two systems with single sources and a common source, both systems including an MPPT, we see that there is a minor improvement of 0.16% for the common source arrangement;
(c) By comparing eqs. (36) and (40), where again there are single sources and a common source, and where the MPPT is included both times, but where the comparison is between individual loads, we see that there is an improvement of 6.78% in the performance of WE1, and a deterioration in the performance of WE2 by 6.62%, in the common source system.

CONCLUSIONS

The paper introduced a procedure for comparing the performances of loads when: the first time they were connected to separate solar cell sources; and the second time when they were connected to a common solar cell source. The comparison of the load performances was also done for the same systems when MPPTs were included. The criterion for the comparison was the "energy utilization efficiency". Two sample loads were analyzed as examples: (a) two different ohmic loads: and (b) two different water electrolyzer loads. The conclusions are:

1. There is an interaction between the loads in a system with a common source, with and without MPPTs, such that one load improves its performance, but at the expense of the other load;
2. In systems not including MPPTs, the total performance of all the loads in a common source system is improved as compared to the performance of all the loads, but when they are powered separately by individual sources;
3. In systems including MPPTs, the total performance of all the loads is the same when powered either by a common source or by single sources;
4. The performance of a load is improved in a system including an MPPT, the amount of the improvement depending on the mismatch of the load to the solar cells;
5. In certain cases, it might be advantageous to disconnect a load from a common source system in order to improve the operation of another load.

REFERENCES

[1] J. Appelbaum, "The Quality of Load Matching in a Direct-Coupling Photovoltaic System," IEEE/PES Winter Meeting, Paper 87 WM 222-3, 1987.

[2] Z. Zinger, A. Braunstein, "Dynamic Matching of a Solar-Electrical (Photovoltaic) System, An Estimation of the Minimum Requirements on the Matching System", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-100, No. 3, pp. 1189-1192, 1981.

[3] C. de Boor, A Practical Guide to SPLINES, Springer-Verlag, N.Y. 1978.

VITA

David C. Schooley

Candidate for the Degree of

Master of Science

Thesis: A STUDY OF PHOTOVOLTAIC ARRAYS UNDER DIVERSE OPERATING CONDITIONS

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in Ft. Worth, Texas, August 7, 1964, the son of Neal and Shirley Schooley.

Education: Graduated from Stillwater High School, Stillwater, Oklahoma, in May 1982; received Bachelor of Science Degree in Electrical Engineering from Oklahoma State University in May, 1987; completed requirements for the Master of Science degree at Oklahoma State University in July, 1989.

Professional Experience: Teaching Assistant and Research Assistant, School of Electrical and Computer Engineering, Oklahoma State University, June, 1987, to December 1988; Worked for the U. S. Army Corps of Engineers, summers of 1985 and 1986.