The trickle charge mode and power dissipation computation explains the inputs I_C, V_C, and V_B to the controller part B. Trickle charge of the battery is thus always being performed with the current I_C. Since I_C will vary with SOC, the shunt controller is comparable to a multi stage controller [nasa].

If the battery is further being charged, the controller remains in the trickle charge mode until the battery is fully charged, and that the upper limit on SOC, F_C specified by the user, is attained. Further charge would result in battery damage and must be restricted. Again a signal, SV (SV = 2), is sent from the controller part A to part B indicating this situation. Controller part B will limit the current to a very small magnitude (0.1 amps) by dissipating the excess power. This current is so small that it does not increase the SOC significantly. Once the battery is fully charged, it can only be charged again without restrictions if SOC drops below F_{CA}, as specified by the user.

When the battery SOC or the battery voltage would exceed the limits on discharge, F_D and V_D, a signal, SV, is sent to the controller part B which sets the battery current equal to zero. In this way the battery is disconnected from the load. This situation has crucial consequences on existing information flow among the components. Since the battery is "turned off", the situation represents a direct coupled system consisting of the PV array and the load and an iteration must be performed between these components to find the operating point of the system. Therefore controller part A sends a signal called signal (signal = 1) to the array and the load component. In the array component, the signal will invoke the convergence promoter which is necessary to run a direct coupled system as discussed in Section 2.6. The effect of the signal in the load component is that the input mode is switched from significant voltage input to significant current input. The iteration procedure (controller A) ignores the battery voltage which would be the open circuit voltage of the battery and takes the voltage from the load, increased by the diode voltage drop, as the output V_{CELL}. The significant input to the load component is now the current received from the cell. The resulting voltage is returned to the controller part A. But before controller part A is called, the controller part B is called from the *TRNSYS* executable program. This controller will receive a battery current equal zero from the add device. Because I_{load} and I_{cell} are identical, I_B is according equation (4.3.2), equal to zero. Therefore the battery will remain disconnected and this operation mode will stay throughout the timestep.

Knowing the battery is discharged and needs to be charged, an effort is made to charge the battery while still matching the load at the following timesteps. At the beginning of the timestep the battery is reconnected and the normal operating mode is used, except that the cell voltage is now the battery voltage plus twice the diode voltage drop. The voltage across the load is also increased to the diode voltage drop, V_{load} which is equal to the sum of V_B and V_{diode}. The controller part B, forced by a signal (SV = 1) received from controller part A, checks on the battery current. If the current turns out to be negative, meaning that the battery would be discharged, the battery current is set equal to zero and a signal, Guard (Guard = 1), is send to controller part A which switches the operation mode back to the direct coupled mode. Otherwise the recent mode will stay until the battery SOC and the battery voltage attain the limits F_{DA} and V_{DA} specified by the user.

To summarize the major decisions, the checks on F and V are done in the controller part A. The controller part B models the valve at the undervoltage protection mode and limits the current at the overvoltage protection mode. Both controllers communicate via control signals.

4.4 MAXIMUM POWER POINT TRACKING

The power output characteristics of a PV cell has a point where the power is at its maximum magnitude. According to Figure 2.3, the maximum power point is close to the knee of the I-V curve of the PV array. Operating the array at this point would guarantee the highest efficiency and the maximum power output of the array. Since this point varies with insolation and temperature, there will be a mismatch between load and array at conditions other than design conditions. Maximum power tracking insures that the array operates at the maximum power point at every instant for any given load.

Many strategies and control schemes have been devised to locate the maximum power point. The most common strategies are based on maximizing the power output of the array. Bucciarelli et al. [28] developed a tracker which is an electronic switching regulator behaving as a dc-to-dc transformer with an adjustable input-to-output voltage ratio. This ratio is determined by a microprocessor-based optimizing controller. The controller continuously maintains PV array operation at the maximum power point by measuring array power and adjusting the ratio using a *hill climbing* algorithm. The maximum power point tracker (MPPT) can operate in four different modes with efficiencies varying from 97% to 99%. Salameh [29] developed a MPPT based on the same strategy, i.e. sensing the power output of the array and searching for the maximum power point. The input-to-output voltage ratio is determined using an analog circuitry based control strategy. A MPPT devised by Landsman [30] differs from those mentioned above but obtains the same effect. The strategy is focused on maximizing the power to the load rather than searching for the maximum power output of the array. This is done by either maximizing the current or voltage to the load. This strategy allowed a simpler technical solution by using digital logic. The efficiency of this MPPT is reported to be 96.7%.

The developed model of a MPPT component incorporates the basic idea of maximum power tracking, that is, to find the input-to-output voltage ratio, N, such that

$$N = \frac{V_{cell}}{V_{load}} \tag{4.4.1}$$

Assuming an ideal MPPT, the currents are related by

$$N = \frac{I_{load}}{I_{cell}} \tag{4.4.2}$$

That means the MPPT is treated as a black box containing basically these equations. Figure 4.12 shows a system configuration including the array, the MPPT and a load.

In reality the MPPT is not ideal and has a certain efficiency. For simplification only the current from the MPPT to the load is reduced by an efficiency factor, that is

$$N = \frac{I_{load}}{I_{cell}\,\eta} \tag{4.4.3}$$

where the efficiency is defined as

$$\eta = \frac{P_{load}}{P_{cell}} \tag{4.4.4}$$

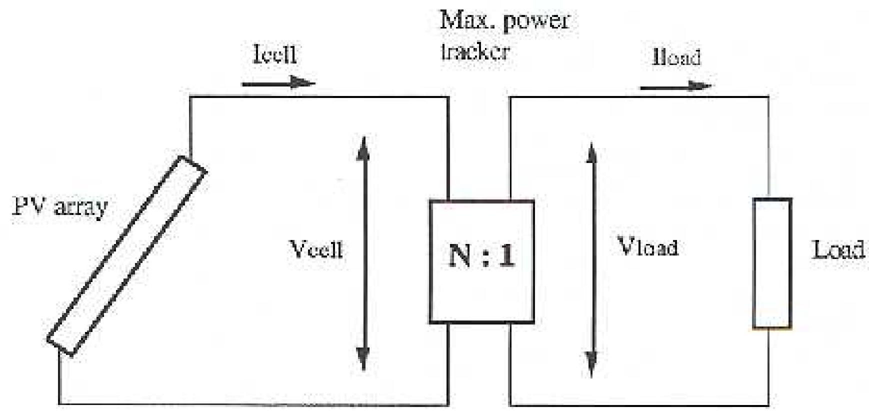

Figure 4.12 Basic maximum power point tracker configuration

The principle of maximum power point tracking is illustrated in Figure 4.13. The original I-V curve modelled a Solarex MSX-30 module at an irradiation of 926.9 W/m^2 and an ambient temperature of 17 °C. As a load, a fixed resistance of 20 ohms was selected. The I-V curve with maximum power tracking was generated applying equations (4.4.1) and (4.4.3) for each I-V pair on the original I-V curve with N = 0.72 and $\eta = 0.97$. The curve with constant power represents the curve at maximum power reduced by the power loss in the MPPT. The maximum power point of the original curve can be found by travelling along the constant power curve until it reaches the point where the power line intersects with the load line (point b). That means that the operating point of the system is now point b). Without maximum power tracking, the system would operate at point a) and thus at a reduced power level.

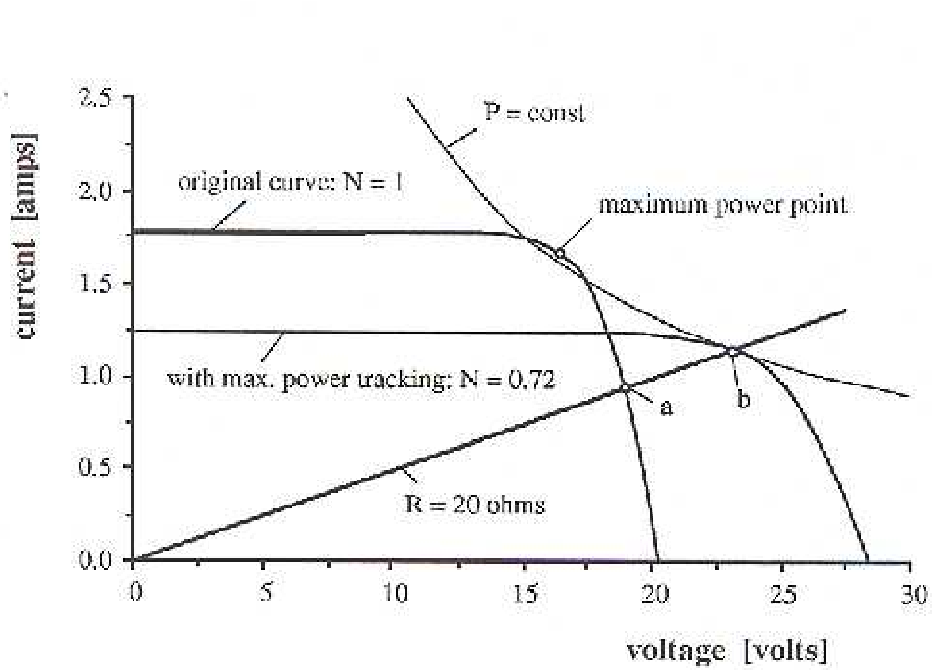

Figure 4.13 Effect of maximum power tracking on I-V curve. The operating points are indicated by (a) and (b).

4.5 TRNSYS COMPONENT OF A MAXIMUM POWER POINT TRACKER

The determination of the maximum power point of the PV array for any environmental condition was already done in the PV array component (see Chapter 2.5). The task for the MPPT component is then simply finding the ratio to adjust the current and voltage at maximum power point to the load condition. The inputs to the component are therefore the current and voltage at maximum power point, I_{MP} and V_{MP} and the load voltage, V_{load}, as shown in Figure 4.14.

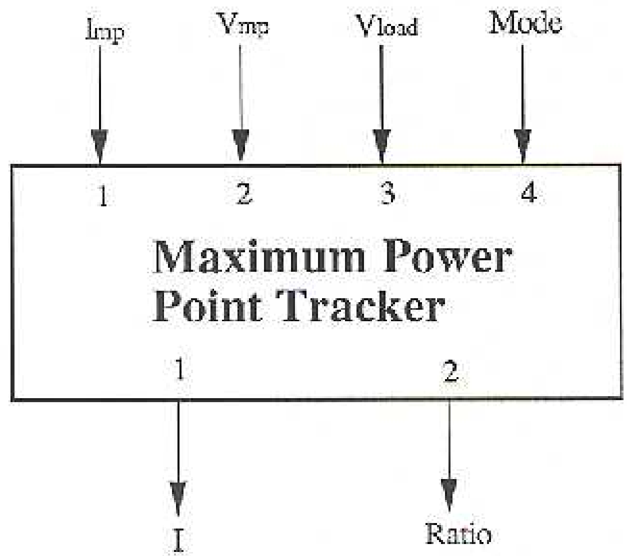

Figure 4.14 Information flow diagram of the maximum power point tracker component

To accelerate the iterative process in finding N, and to assure convergence, a convergence promoting algorithm is included in the component. The convergence promoter is used when running a system without battery storage. Since the operating voltage is within a small range as determined by the battery voltage when running a system with battery, a solution is found after a few iterations using successive substitution. The input Mode fixes the desired operation mode. Mode equal to zero means that the convergence promoter is turned off otherwise it is turned on.

The only parameter of the component is the MPPT efficiency, which is assumed to be constant over the entire operating range. The outputs are the current for use in the load component and the input-to-output voltage ratio, N.

Chapter 5

SYSTEM ANALYSIS

Chapters two, three and four treated the mathematical models of the individual components of a PV system. The theory of these models were discussed and I-V curves describing their electrical behavior were generated. This chapter treats the characteristics of the entire PV systems. The intention is to discuss the behavior of PV systems in general and to point out some typical features. It is not intended to provide detailed parametric studies or optimization studies. It is demonstrated how information flow diagrams of the PV systems can be constructed to run *TRNSYS* simulations. Hints on how to use *TRNSYS* are given. Firstly, direct-coupled systems are discussed using motor loads as representative examples. Secondly, systems with battery storage are examined. Finally the effect of maximum power point tracking on both kinds of systems is studied.

5.1 DIRECT COUPLED SYSTEMS

PV systems, where loads are directly coupled to the PV array, i.e. without power conditioning devices or battery storage, are called direct-coupled systems. These systems are in use for cases where the load profile follows the solar radiation profile.

The most common application are water pumping systems. Irrigation of farmland can be done during the day when solar energy is available.

Two motor load applications are examined: a motor ventilator load and a motor centrifugal pump load. The two developed DC-motor models are compared by simulation and evaluated.

Motor-ventilator load

The behavior of direct-coupled systems can be examined by superimposing the PV array I-V curves on the load I-V curves. For the same ventilator-motor combinations, presented in Chapters 3.3.1, 3.3.2, and 3.3.4, the I-V curves are shown in Figure 5.1.

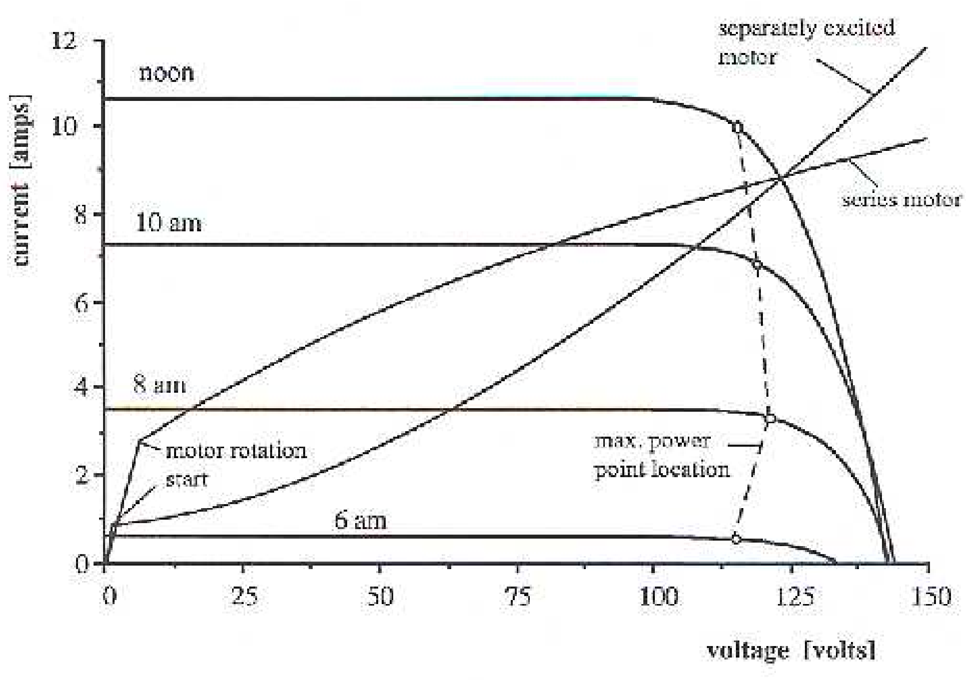

Figure 5.1 I-V characteristics of a PV generator and DC motor-fan loads (separateley excited and series motor); Solarex MSX-30 array (7 modules in series, 6 modules in parallel); conditions are those in Chapter 3

The PV array I-V curves correspond to the day time conditions as indicated. These are the same conditions at which the motor n-T curves (Figures 3.3 and 3.5) were conducted. The data of the array, the motors, and the ventilator are listed in Appendix C. The system operates where the array curves and the load curves intersect. The separately excited motor starts rotation earlier than the series motor. The task of system design is to operate the load close to the maximum power points of the PV array. Although both motor curves do not follow the line of the maximum power points of the PV array curves, the separately excited motor behaves better. Most of the time, the operating points of this motor are much closer to the maximum power point than the series motor operating points. This is especially true at low radiation levels.

The system can be optimized by selecting the number of modules in series and in parallel so that the load matches the maximum power points of the array most often throughout the year. This is a difficult task because the location of the power points depend on temperature and irradiation varying with time. However, one design criteria is to match the maximum power point at an irradiance of approximately 800 W/m^2.

Motor-pump load

The electrical behavior of the motor-centrifugal pump load is similar to the motor-ventilator load. In Figure 5.2, the I-V characteristics of a PV array and the motor pump loads are shown. The data for the series and separately excited motor and the pump are the same as used in Chapter three. The PV array I-V curves are obtained at a constant cell temperature of 25 °C and at five different insolation levels.

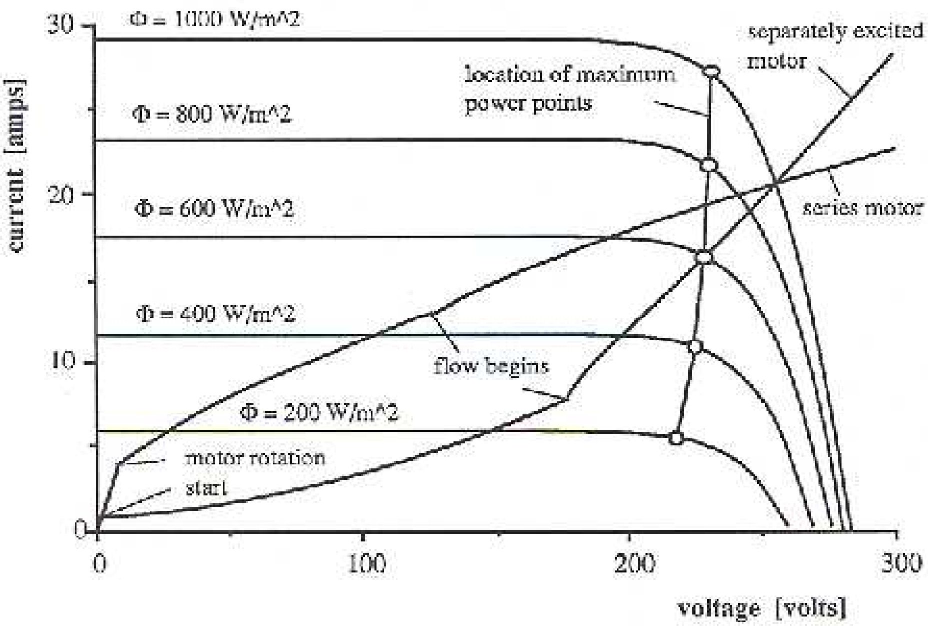

Figure 5.2 I-V characteristics of a PV generator and a centrifugal pump driven by a separately excited and a series motor. Kyocera module (14 modules in series, 10 modules in parallel); Tcell = 25°C at five insolation levels

The separately excited motor is more compatible with the array than the series motor. The power requirements and hence the irradiance to start the motor and to provide flow are less for the separately excited motor than for the series motor. The power required to deliver water depends on the static head of the pumping system; less static head requires less power.

Both motors are compared by running a daily and a yearly simulation at Madison, Wisconsin. The daily simulation was done for the first of June using *TMY* weather data. The delivered water versus the time of the day for both motor combinations is shown in Figure 5.3. The *TRNSYS* decks, containing the data of the system components, are listed in Appendix B.

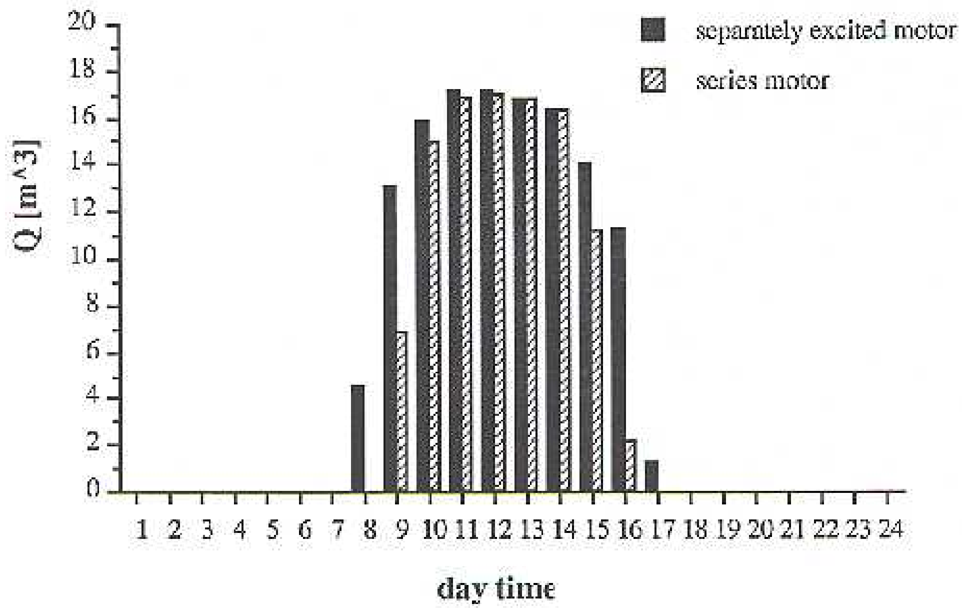

Figure 5.3 Daily volume of pumped water for two different motor-pump combinations. In Madison, Wi, on June 1st.

As it is expected, the separately excited motor delivers more water in the early morning and the late afternoon. The difference is not significant in the middle of the day where the motors supply approximately the same flowrate. Considering the entire day, the separately excited motor delivered approximately 20% more water than the series motor.

Regarding a yearly period, the trend is the same. Figure 5.4 shows the flowrate versus time graph for the *TMY* year in Madison, Wi. The water pumped using the separately excited motor is about 24% more than the water pumped using the series motor.

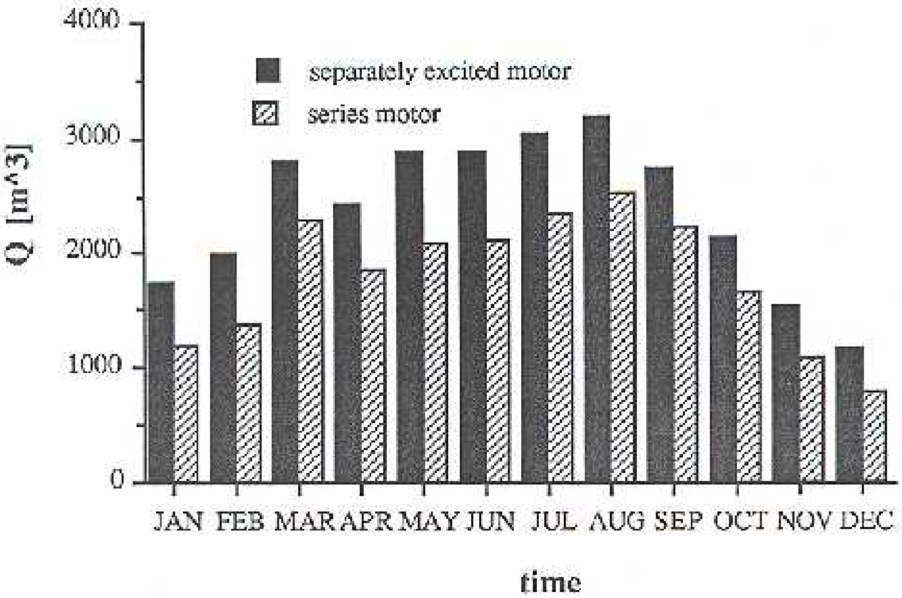

Figure 5.4 Yearly volume of pumped water for two different moto-pump combinations
Location: Madison, Wi

5.2 SYSTEMS INCLUDING BATTERY STORAGE

This section considers systems where the battery is connected in parallel to the PV array and to the load as shown in Figures 4.4 and 4.8. The operating voltage of such a system depends on the I-V characteristics of all involved components, that is the PV array, the battery, and the load. Since all components are wired in parallel, the voltage across their terminals must be the same for all components (neglecting the voltage drop across the diode). The operating voltage is determined at the voltage where Kirchhoff's current law is satisfied. There is only one voltage where this is the case. Depending on the insolation level, the battery is on charge or discharge for a given load profile. If the irradiance is large enough, excess energy can be used to charge the battery. Figure 5.5 illustrates the operating conditions for a system with battery on charge.

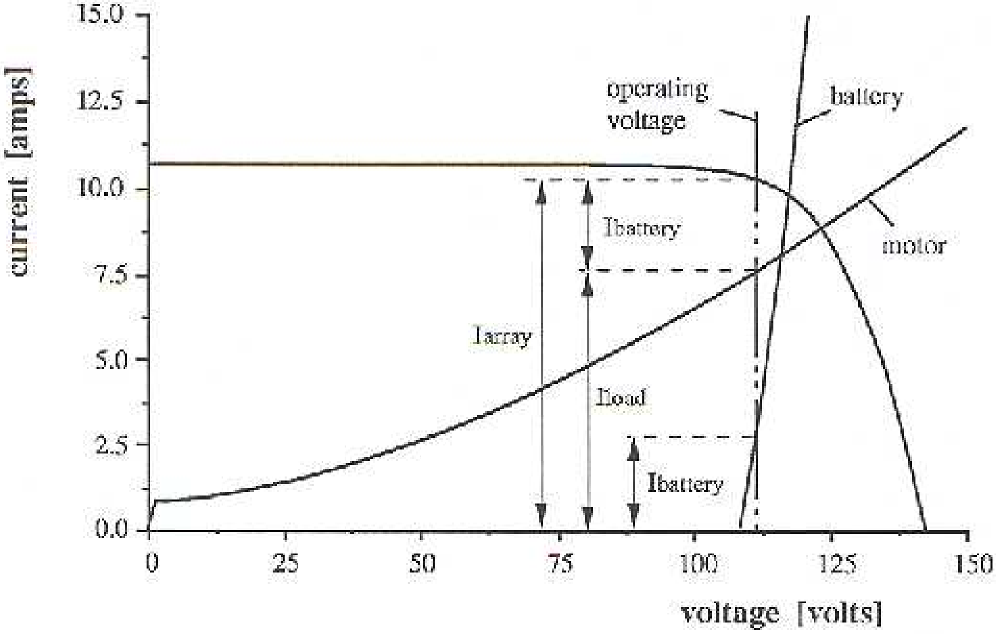

Figure 5.5 Illustration of the operation of a PV system including PV array, battery on charge and a separately excited motor-fan load. Battery: lead acid; rated 250 Ah, 50 cells in series, SOC = 100%. PV module: Solarex MSX-30; 7 modules in series, 6 modules in parallel, at 963 W/m^2 and ambient T = 16.7°C

The operating voltage is given at the voltage where the difference between the array current and the load current is exactly equal to the current into the battery.

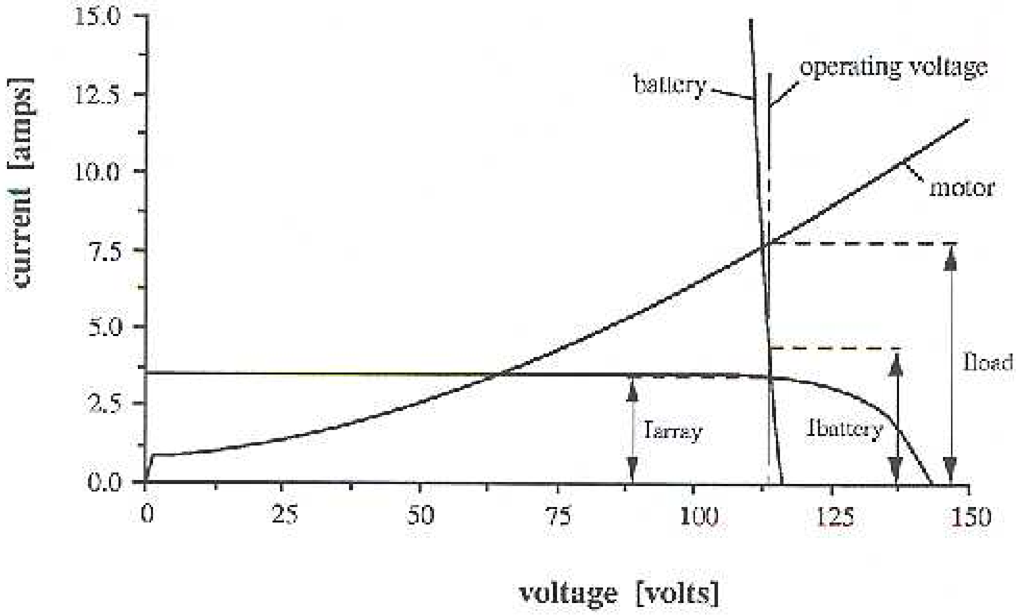

Figure 5.6 Illustration of the operation of a PV system including PV array, battery on discharge and a separately excited motor-fan load. Battery: lead acid; rated 250 Ah, 50 cells in series, SOC = 100%. PV module: Solarex MSX-30; 7 modules in series, 6 modules in parallel, at 319 W/m^2 and ambient T = 13.9°C

The situation for discharge is shown in Figure 5.6. The battery I-V curve is drawn in the first quadrant for better illustration. Corresponding to the chosen current sign convention for the battery component in Chapter four, the battery I-V curve would be located in the fourth quadrant. Figure 5.6 shows the situation where the array current cannot match the load within the voltage range of the battery. To make up the difference between the load and the array current, the battery current is necessary. The operating voltage is determined where the load current is equal to the sum of array and battery current.

For a given fixed motor I-V curve, the operating voltage is nearly determined by the battery terminal voltage. The battery acts like a voltage regulator and can be used to operate the PV array close to its maximum power points. This can be accomplished by selecting the proper number of battery cells in series so that the battery I-V curves match the maximum power line of the PV array, as shown in Figure 5.7.

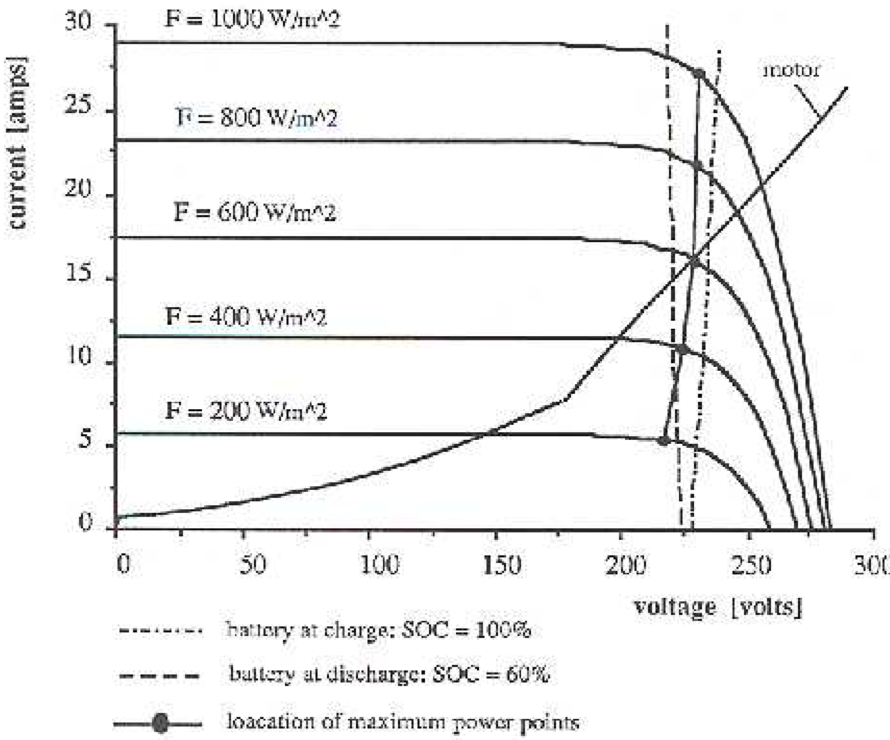

Figure 5.7 I-V characteristics of a PV generator, a lead acid battery (charge and discharge), and a separately excited motor-centrifugal pump. Kyocera module: 14 modules in series, 10 modules in parallel. Battery: rated 250 Ah, 105 cells in series, 5 cells in parallel. Cell temperature = 25°C

The operating range of the system is determined by the largest battery voltage, that is when the battery voltage is fully charged and the gassing voltage is reached, and by the smallest voltage permitted, that is the cutoff voltage on discharge. The I-V characteristics of the battery vary with SOC and the battery voltage depends on the charge or discharge current. On the other hand, the maximum power line changes with radiation and temperature. Therefore an exact match between battery and load can generally not be attained. A proper design, however, of the entire system may result in a good utilization of the PV array.

If storage is necessary, water storage is commonly used for water pumping systems. Battery storage might be appropriate when pumping is required in large quantities or when pumping is required by a fixed quantity per hour or per day. A battery can improve system performance at large system static heads requiring a high insolation level to provide flow. The battery can be designed to operate above the knee of the pumping system I-V curve, i.e above the point where flow begins. This guarantees a continuous flow.

The performance of a water pumping system with battery storage is shown in Figures 5.8, 5.9, and 5.10. The daily solar radiation variation and duty cycles of the battery for 48 hours are shown in Figure 5.8.

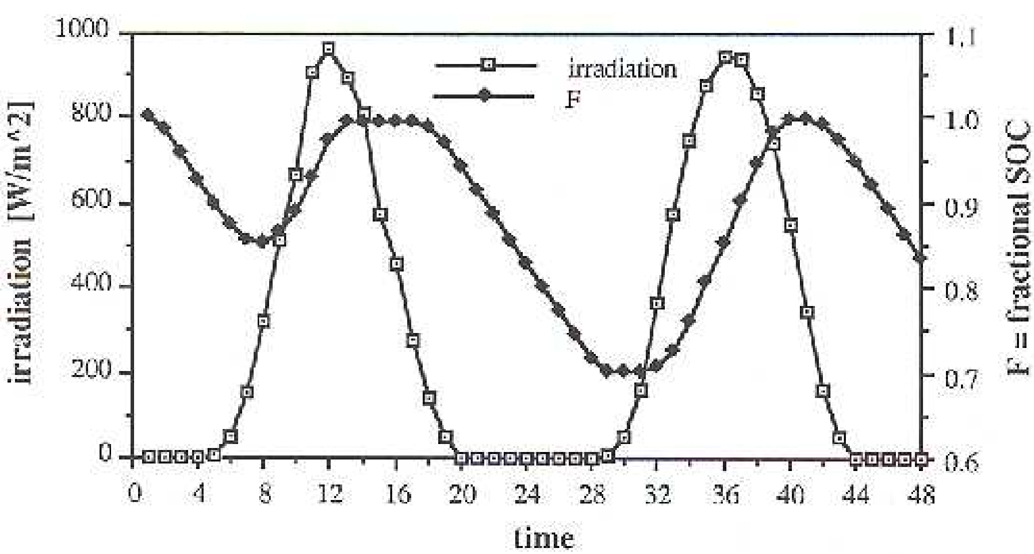

Figure 5.8 Solar radiation and SOC of battery curves

The battery was permitted to operate between full charge (F = 1) and 70% SOC (F = 0.7) using the shunt type charge controller. The battery SOC follows the variation of irradiance. Starting at 1am, the battery is being discharged until the radiation level is high enough to charge the battery. At hour 13, the battery is fully charged and the excess energy is dumped.

In Figures 5.9 and 5.10, the performance of the system is illustrated comparing the series and the shunt type charge controller. The pumped water versus time is plotted in Figure 5.9. The amount of pumped water per hour changes a little and is within the range of approximately 13 to 17 cubicmeters per hour, except at hours from 27 to 31. At that time the battery reaches the discharge limit. In this case, the series controller disconnects the motor from the battery and the battery is on charge mode. At hour 32, the battery is charged enough to reconnect the load. When using the shunt controller, the battery reaches the discharge limit at hour 28. In contrast to the series controller, the shunt controller disconnects the battery from the load while it is still attempted to match the load. At this operating stage, however, there is only enough solar energy available to provide flow at hour 31. At hour 32, the radiation level permits matching the load and charging the battery again.

Figure 5.10 is a plot showing the battery fractional SOC versus time of both charge controllers. The advantage of the modeled charge controllers is obvious regarding the control strategy when the battery becomes fully charged. The series controller disconnects the array from the battery and the battery is discharged by the load. On the other hand, the shunt controller is still connected to the system bus and stays on charge receiving a small charge current. The result is that the battery stays on a higher charge level than is the case for the series controller which has also the effect that the

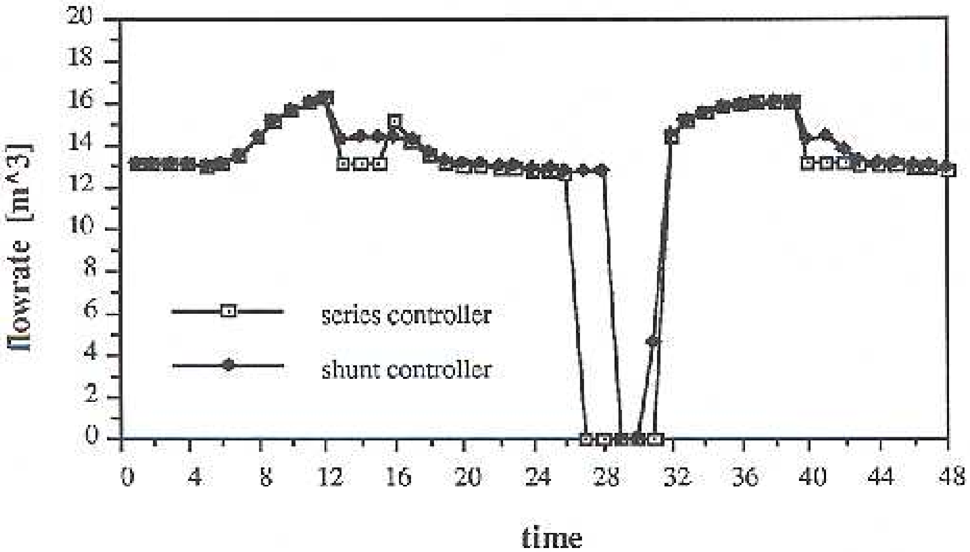

Figure 5.9 Volume of flowrate versus time for the series type and shunt type charge controller

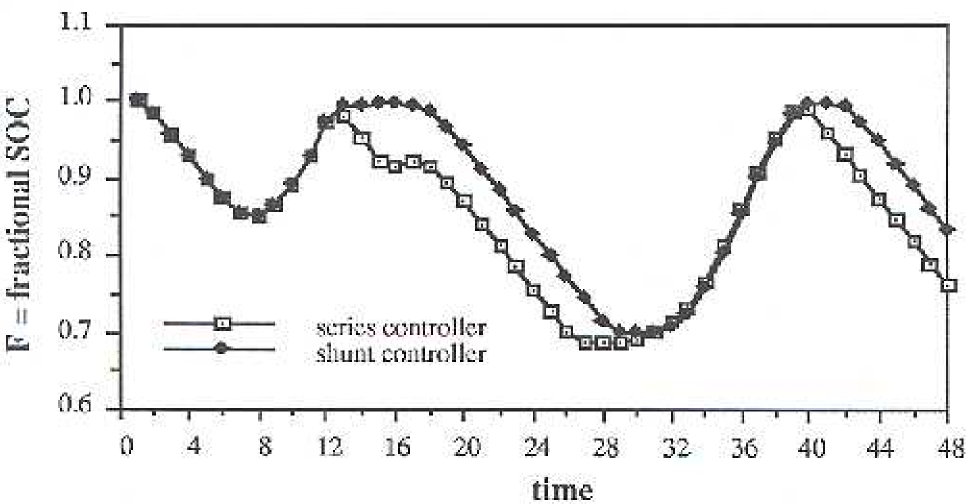

Figure 5.10 Battery SOC versus time for the series type and the shunt type controller

operating voltage is higher and hence the pump provides more flow (hours 13 to 15). The jump in flowrate and F at hour 17 is due to the fact that the battery SOC is discharged below the deadband limit after being fully charged and is then on charge again.

5.3 SYSTEMS WITH MAXIMUM POWER TRACKING

The performance of some direct-coupled PV systems can be improved immensely using a MPPT. Because of the mismatch between the load I-V curve and the array I-V curve discussed in Section 5.1, the maximum power tracker will assure that the PV array always operates at its maximum power point. The results from a comparison of a water pumping system including a MPPT and the direct-coupled system is shown in Figure 5.11. A separately excited motor was used to drive the centrifugal pump. The maximum power tracked system provided a total of 8.3% more water than the direct-coupled system for the 1st of June using *TMY* data. The MPPT efficiency was assumed to be 97% efficiency and the system data are listed in Appendix C. A printout of the *TRNSYS* deck is listed in Appendix B.

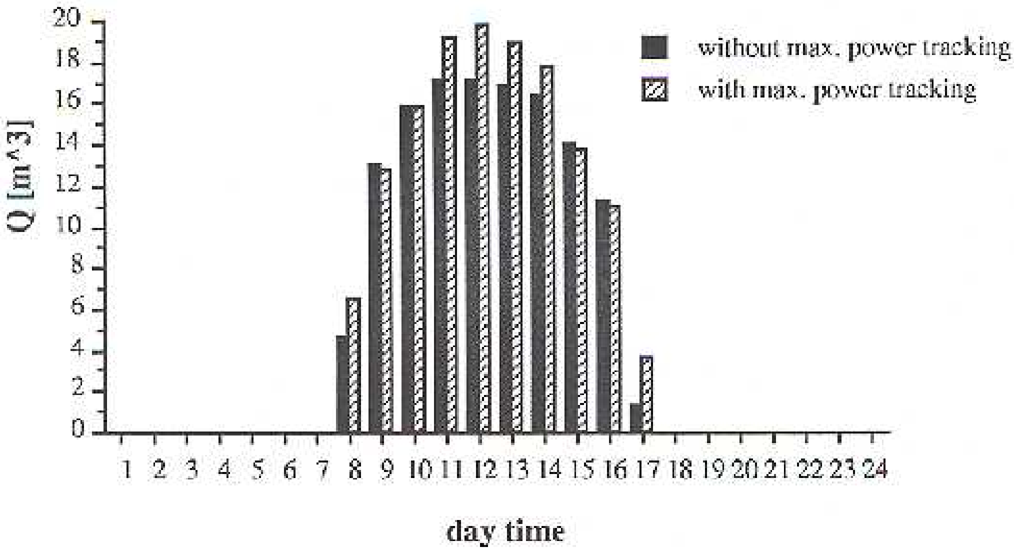

Figure 5.11 Flowrate versus time for a direct-coupled motor-pump load and with maximum power tracking. Location: Madison, Wi.

Using a MPPT in systems with battery storage, the effectiveness is rather low. This is due to the fact that such a system, if designed properly, already operates close to the maximum power points of the array. Improvement requires a MPPT with a very high efficiency. Figure 5.12 shows the results for a yearly simulation of a battery system with and without MPPT. The system with MPPT yielded only an increase of approximately 1% pumped water. The simulation provides rather an ideal situation. The temperature effects on the battery performance is not considered in the battery model. For example, a battery exposed to the environment could experience, depending on the site, a large temperature variation. In reality the mismatch of the load, battery and the PV characteristics is larger than the simulated one and a MPPT might be more effective.

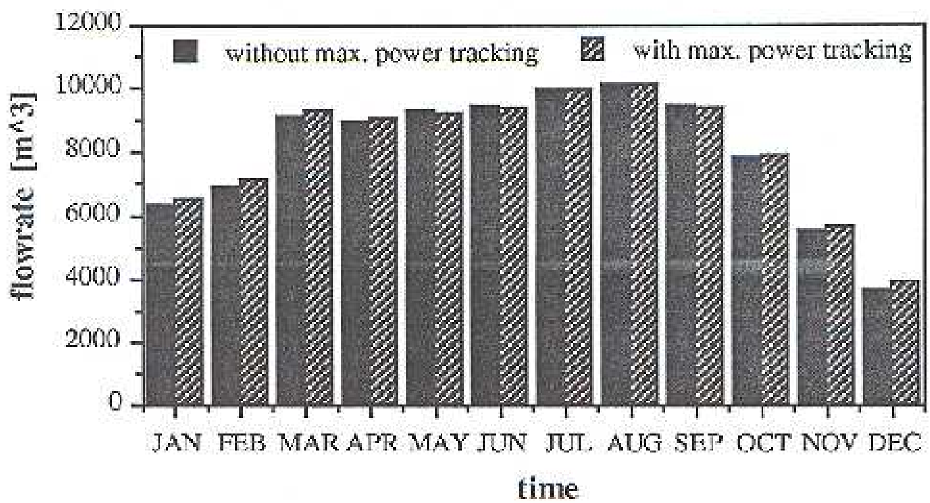

Figure 5.12 Results from yearly simulations of a system with battery storage, with and without maximum power tracking

5.4 TRNSYS System Information Flow Diagrams

A brief introduction of *TRNSYS* has been given in Chapter one, and some comments about how a *TRNSYS* deck has to be constructed to run a simulation has been made in Chapter four. In this section some additional explanations and suggestions are presented.

All the components of the system and the information flow between the components are specified in the deck. The information flow can be visualized drawing the information flow diagram. As an representative example for direct-coupled systems, a system including a PV array, a motor, and a fan are chosen. The information flow diagram of this system is shown in Figure 5.13. Only the important inputs describing the information flow are shown. Not shown is the data reader component and the solar radiation processor component. The data reader is used to make the meteorological data available to the simulation. Because the solar radiation is usually given for a horizontal surface at hourly intervals, the solar radiation processor may be used to estimate insolation on up to four surfaces of either fixed or variable orientation, according to established methods [2],[31]. The solar radiation processor also interpolates the data for timesteps shorter than the timestep for which data are available. The first input of the PV array is the irradiation which is taken from the solar radiation processor. The ambient temperature is taken directly from the data reader.

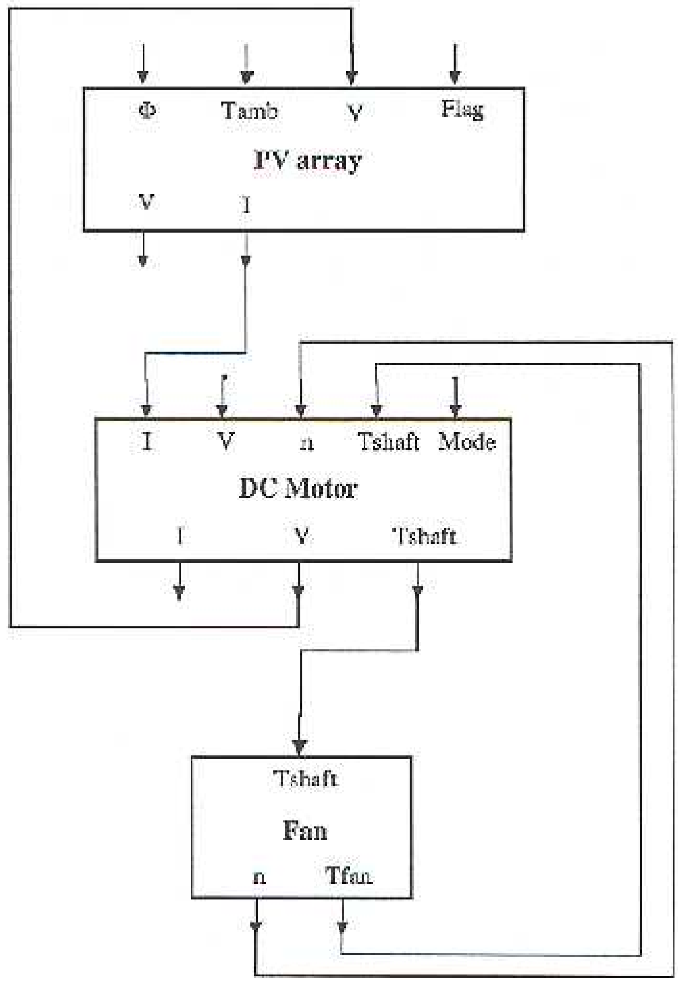

Figure 5.13 TRNSYS system configuration of a PV system including a PV array, a DC motor and a fan

Irradiation and temperature are the forcing variables for the simulation. For a given set of irradiance and temperature, the *TRNSYS* executable program calls the individual components sequentially and performs the previously described iterative process to find the solution. A solution is found when the change of all inputs from the previous and the current iteration is less than a specified tolerance. *TRNSYS* then moves on to the next timestep and repeats this procedure for a different set of irradiance and temperature.

The calling sequence of the individual components has to be specified in the *TRNSYS* deck. In the case of the system in Figure 5.13 the calling sequence is as follows: the PV array is called first and the current is evaluated given an initial voltage guess and the weather data. The current is transferred to the motor component. As already explained in Chapter 3.3.3, a recyclic subloop between the motor and the fan is involved. This subloop can be forced by using the *TRNSYS loop* command (see sample decks in Appendix B). The motor and the fan are called until the speed, the torque, and the voltage, corresponding to the current from the array, are determined. The voltage is returned to the array and provides a new guess resulting in a new current.

TRNSYS provides a tool called the accelerator, which promotes convergence and accelerates the described iterative process. This accelerator should not be used in simulating the PV systems. Convergence promoting algorithms are included in the PV array component and the motor component; an additional convergence promoter would interfere and might lead to erroneous results.

Replacing the fan by the hydraulic system component allows the simulation of the direct-coupled water pumping system. In the case of a resistance load, the motor fan components are simply replaced by the resistance component. The information flow diagram for such a system is simpler because the subloop can be omitted.

Information flow diagrams for systems with battery storage were shown and explained in Chapter four. It remains to be said that if simulating motor loads, the load component shown in Figures 4.7 and 4.11 can be replaced by the motor and pump component.

A MPPT can be included and connected as it shown in Figure 5.14. The voltage from any load is now input to the MPPT, instead to the PV array, and the current to the load has to be taken from the MPPT. For a system with battery storage, the voltage from the charge controller would be input to the MPPT.

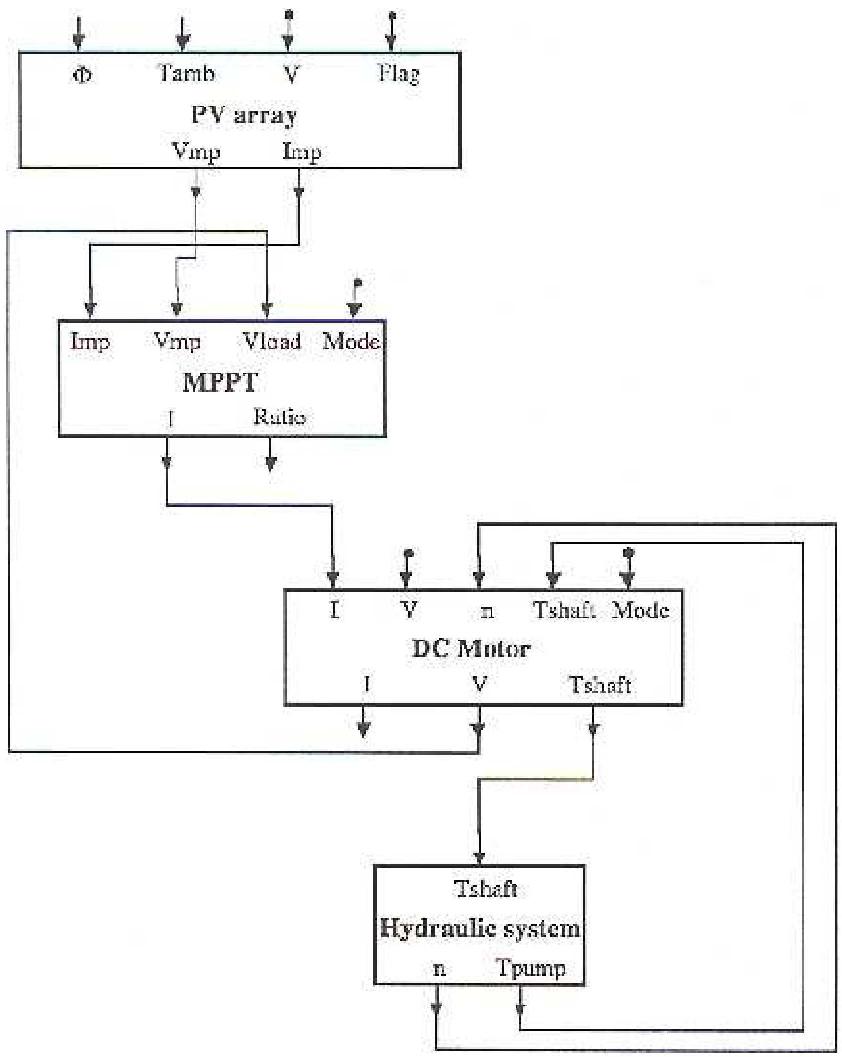

Figure 5.14 TRNSYS system configuration of a PV system including a PV array, a DC motor, a pump, and a maximum power tracker.

Finally, some suggestions about the convergence tolerance and the number of iterations permitted to find the solution are presented bellow. The tolerance can be specified by the user. The accuracy of the simulation and the length of the simulation, i.e. the number of iterations required to find the operating point depends on the chosen tolerance. For all simulations performed using the developed PV system components, a relative tolerance of 0.001 provided a satisfactory tradeoff between accuracy and amount of iterations and is recommended. The number of allowed iterations also can be specified by the user. Simulating direct-coupled systems with a resistive load, a limit of 30 iterations was never exceeded. For motor loads this number has to be increased, because of the involved subloop. A total number of about 60 iterations is suggested and the motor subloop should be allowed to iterate 30 times. The limit has to be increased again when simulating battery systems. Systems including the shunt type controller requires more iterations because control decisions made during the timestep might completly change the operating situation of the system. For instance, if the battery reaches the discharge limit, the controller disconnects the battery from the load, which is then equivalent to a system without battery, i.e. the operating point has to be determined between array and load. A limit of 200 iterations is proposed for these systems. Care must be taken in setting the charge controller limits. If the deadband limits are not chosen properly, the controller might oscillate and TRNSYS may not converge.

Chapter 6

SIMULATION AND COMPARISON

To validate the developed computer models, a comparison between *TRNSYS* simulations and experimental data has been made. Simulations of a direct-coupled water pumping system and a water pumping system with maximum power tracking are compared to experiments conducted at the Florida Solar Energy Center (FSEC) [32].

The data are from a series of experiments of different system configurations conducted at FSEC. Two system configurations were selected and compared with computer simulations. System A consists of a centrifugal pump driven by a 1/2 hp, 36 volt DC permanent magnet motor operated from a single-axis passive sun tracking array. System B consists of the same centrifugal pump-motor combination operated from a fixed array coupled to a MPPT.

The arrays in both system configurations have the same maximum power rating, but are configurated differently. For system A, three modules are connected in series and two modules are connected in parallel. For System B, all six modules are wired in series. The head versus flow rate characteristics of the pump-motor combination were given for a range of operating voltages. In order to run the simulations, the pump performance data had to be made compatible with the requirements of the *TRNSYS* component of a hydraulic system. The required data for this component are head-flow rate and efficiency-flow rate data pairs at a chosen reference speed. The motor parameters were also not given explicitly. The series resistance could be obtained from the slope of the pump-motor I-V curve and the motor constant could be obtained applying the motor relations presented in Chapter 3 to the rated motor values.

The system head-flow rate profile is shown in Figure 6.1. The system head consists only of a dynamic head, i.e. the water was just circulated through a circuit. The Figure shows the measured system head profile and a fitted curve approximating the measured data according to equation (3.3.16). The constants of this equations were determined to be $k_1 = 0.118$ and $k_2 = 4.9e6$. The nature of the polynomial described by equation (3.3.16) does not allow the intersection with the Q axis if the head has to be restricted to positive values. Hence, the fitted curve does not match accurately at the region where flow begins. The difference between the curves becomes significant at flow rates greater than 0.0012 m^3. Since the pump was operated below this point, the curve fit is a good approximation to the measured curve.

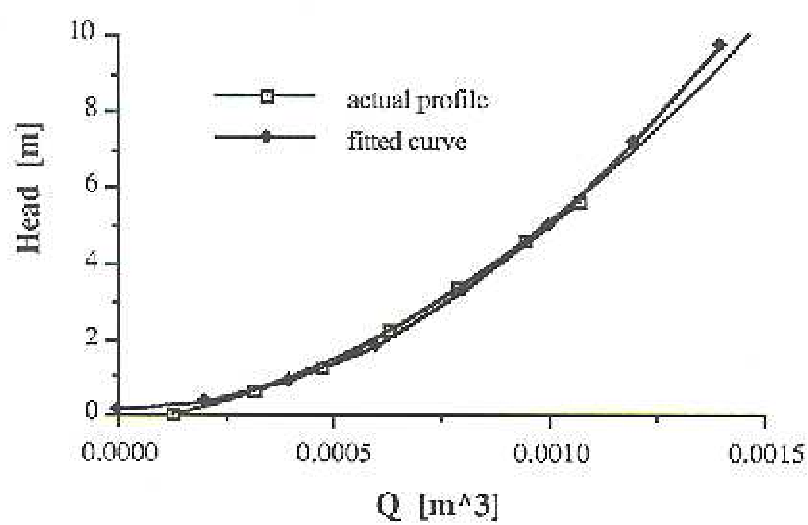

Figure 6.1 System head profile. The measured and the fitted profiles are shown

The experiments were conducted over a period of one day. Figure 6.2 shows the irradiance profiles of the two different systems at that day. The ambient temperatures were not available. It is assumed a temperature range from 26°C in the morning to 32°C at noon, typical for the location (Cape Canaveral, Fl) and the season of the year (August) at which the experiments were conducted.

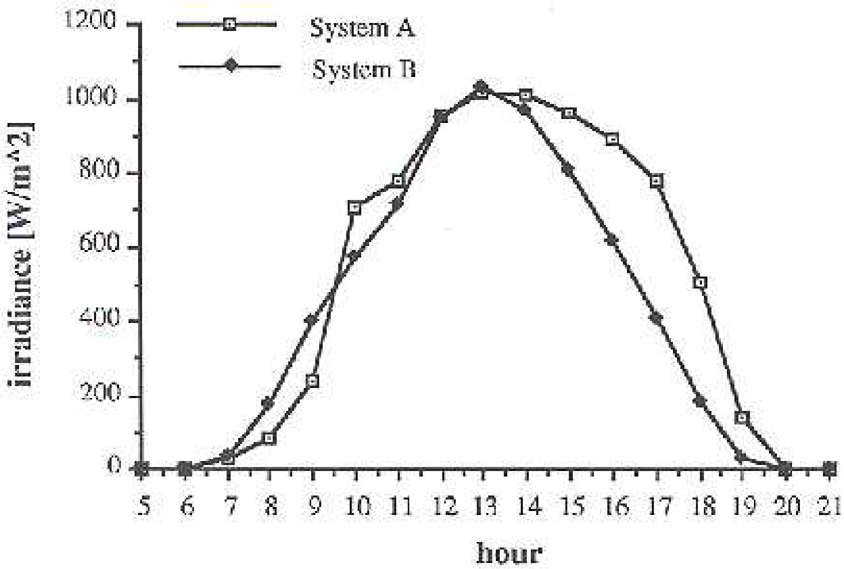

Figure 6.2 Daily irradiance profiles of the simulated systems

Additional assumptions had to be made about the array parameters, μ_{ISC}, μ_{VOC}, NCS, T_{CNOCT}, T_{ANOCT}, Φ_{NOCT}, and the static friction and the viscous friction components, C_{stat} and C_{visc} of the motor torque loss relation. All the data of the system components as used in the simulations are listed in Appendix D.

Strategy of the Simulations

First System A is simulated using the data and assumptions stated above. Since the uncertainty involved in reading the data from the given graphs and the inaccuracy involved in assuming the parameters which were not available, a parametric study is conducted to investigate the effect of the assumed parameters on the performance of the system. Once the trends of the simulations in comparison to the experimental data are known for System A, System B can be studied and the performance of the modelled MPPT in comparison to the real device can be evaluated.

System A

A system can be examined looking at the performance of its individual components and looking at its overall performance. At first the PV array is examined comparing the measured and the predicted maximum power output. Then the entire system is examined studying flow rate , head and the pump-motor efficiency.

The model of the PV array has already been proven to simulate the I-V characteristics accurately [3]. The maximum power point is an appropriate parameter to evaluate the model, since maximum power output is independent of the connected load. Figure 6.3 shows the measured and the predicted maximum power output during the day. There is almost no difference between the predicted and the measured values in the morning and the late afternoon. The difference is significant between hours 13 to 15 and was a maximum of 4.5% at hour 13. The simulation provided, however, rather conservative results. The mismatch at the hours with large radiation and high temperatures might be due to an incorrect assumption of the NOCT temperature or the actual ambient temperature. An effect of a change of the assumed parameters μ_{ISC} and μ_{VOC} could not be observed as it is shown in Figure 6.4.

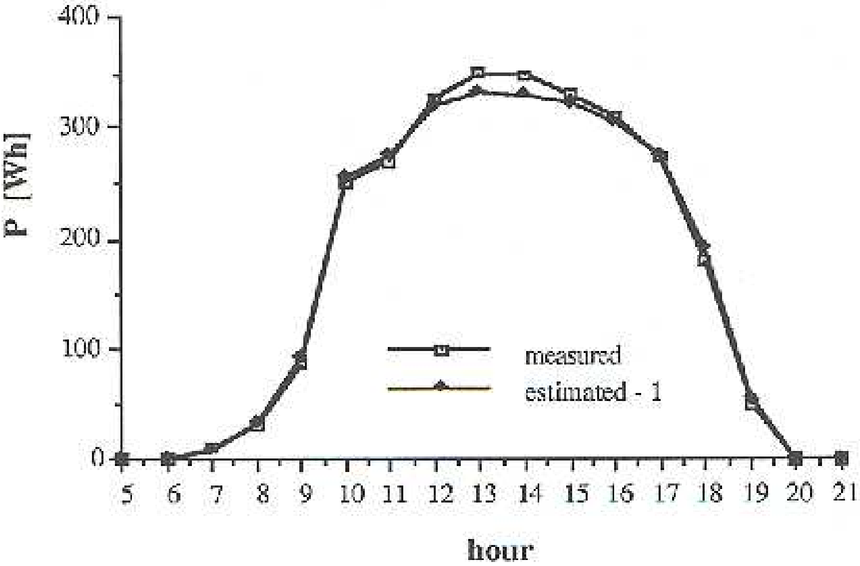

Figure 6.3 Maximum power output of the array. The measured and the predicted curves are shown

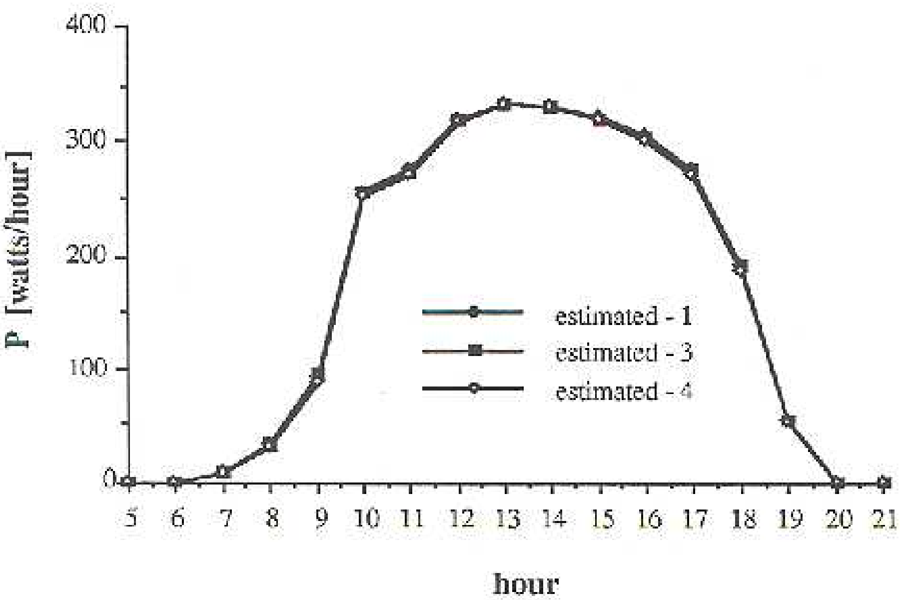

Figure 6.4 Maximum power output of the array. Estimated -1 is the original curve. Estimated - 3 μ_{isc} = 1.5e-3 A/K and μ_{voc} = -75.5e-3 V/K. Estimated - 4 μ_{isc} = 3.e-3 A/K and μ_{voc} = -79.5e-3 V/K

The performance of the entire system is illustrated in Figures 6.5-7. The simulations were run using two sets of motor parameters. Thus the eventual uncertainty involved in reading the data from the graphs can be explained. The shape of the measured as well as the shape of the predicted curve are smiliar. At the first hour and the last hour where solar radiation is incident, the simulations yield larger quantities of pumped water than the experiment. This might be due to the mismatch of the measured and the fitted system head-flow rate profile at low system head, as shown in Figure 6.1. The effect of the motor parameters is apparent. The curve indicated as *estimated-1* was simulated using a series resistance of 0.43 ohms and a motor constant of 0.188 Vs obtained from the given data. For the curve indicated as *estimated-2*, a series resistance of 0.59 ohms and a motor constant of 0.173 Vs was used. The different motor constants result also in different pump head-flow rate and head-efficiency characteristics, since the data were given in a form where the motor and the pump were considered as one unit. Increasing the internal resistance decreases the motor pump performance. The daily difference of pumped water between the measured data and the simulated data of *estimated-1* is approximately 2.5% while the largest difference is 6% observed at hour 13. The simulated curves provided rather conservative results. The same trends can be observed looking at the total head developed during the day and the daily motor-pump efficiencies as shown in Figure 6.6 and 6.7. Increasing the series resistance of the motor causes a shift of the motor-pump I-V curve towards the I axis, i.e. away from the maximum power point of the array. The system is then operated within a region of reduced motor-pump efficiency and therefore less hydraulic power can be developed , i.e. less flow and less head. Thus the conclusion can be drawn that the mismatch between the measured flow and the simulated flow (*estimated-1*) can be due to the use of an inaccurate value of the motor parameters. Another factor might be that the PV array model predicted less maximum power than the real array. However, the results of the simulated performance of the water pumping system is quite good. As mentioned above the predicted daily flow was 2.5% less than the measured flow.

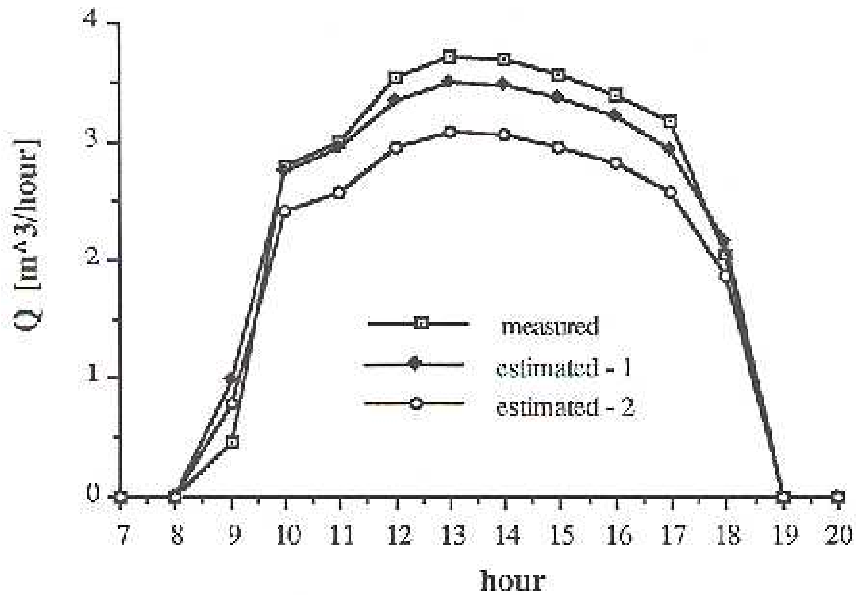

Figure 6.5 Daily flow rate measured and estimated. Estimated - 1: Ra = 0.43 ohms, kφ = 0.188 Vs; Estimated - 2: Ra = 0.59 ohms, kφ = 0.173 Vs

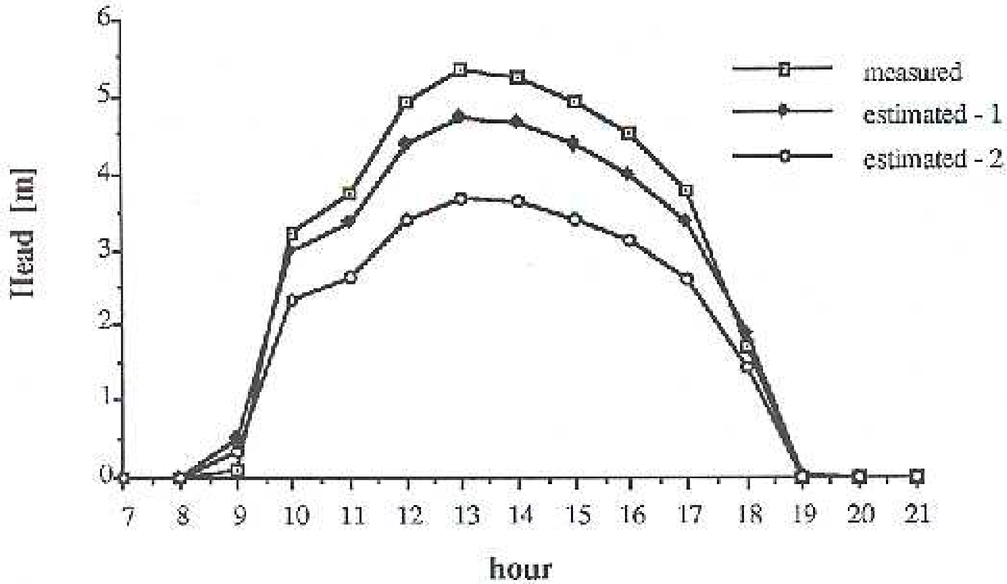

Figure 6.6 Developed total head, measured and estimated. Estimated - 1: Ra = 0.43 ohms, kϕ = 0.188 Vs. Estimated - 2: Ra = 0.59 ohms, kϕ = 0.173 Vs

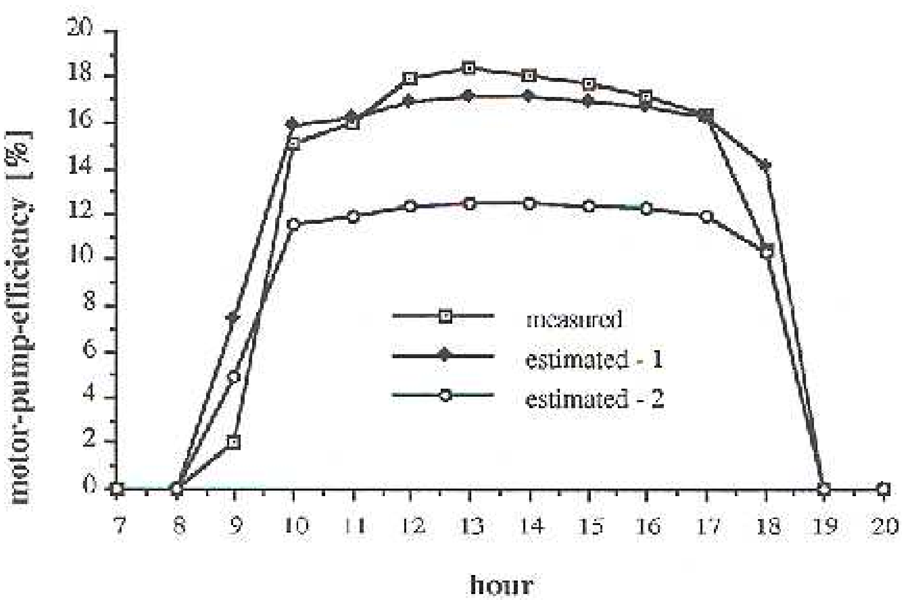

Figure 6.7 Motor-pump efficiency, measured and estimated. Estimated - 1: Ra = 0.43 ohms, kϕ = 0.188 Vs. Estimated - 2: Ra = 0.59 ohms, kϕ = 0.173 Vs

System B

The only difference in the system configuration to the system above is the addition of a MPPT. Therefore the analysis of this system is concentrated on evaluating the developed MPPT model.

The flow rate versus time of the day curves for the real system and the simulated system are shown in Figure 6.8. The shape of both curves is again of the same quality. The MPPT model predicted higher flow rates than the real system except for the time from noon to 3pm. The reason for that can be found looking at the MPPT efficiency of the real system as shown in Figure 6.9. At low insolation levels, the efficiency of the real MPPT drops from about 93% at high insolation levels drastically to approximately 35%. At low insolation levels, the intersection of the load I-V curve with the array I-V curve is further away from the maximum power point of the array than at high insolation levels. The input-to-output voltage ratio is in this case much greater than one, which a real MPPT cannot handle without sacrificing efficiency. The efficiency of the MPPT model was assumed to be 93% over the entire voltage range. The MPPT model overestimates the performance of the system because of its ideally behavior. The predicted flow rate is 9.7% higher than the measured flow rate.

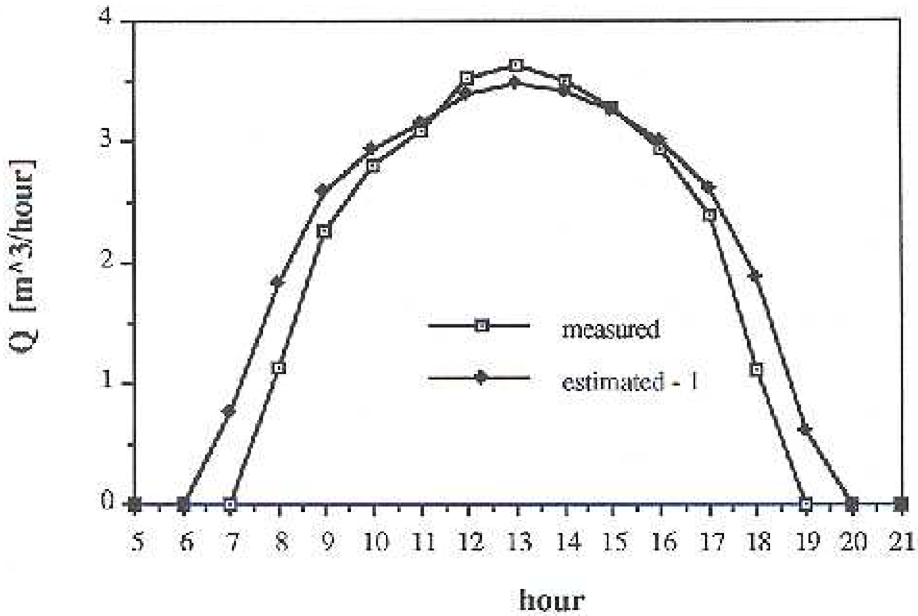

Figure 6.8 Flow rate versus time for a system with MPPT. Measured and predicted performance curves

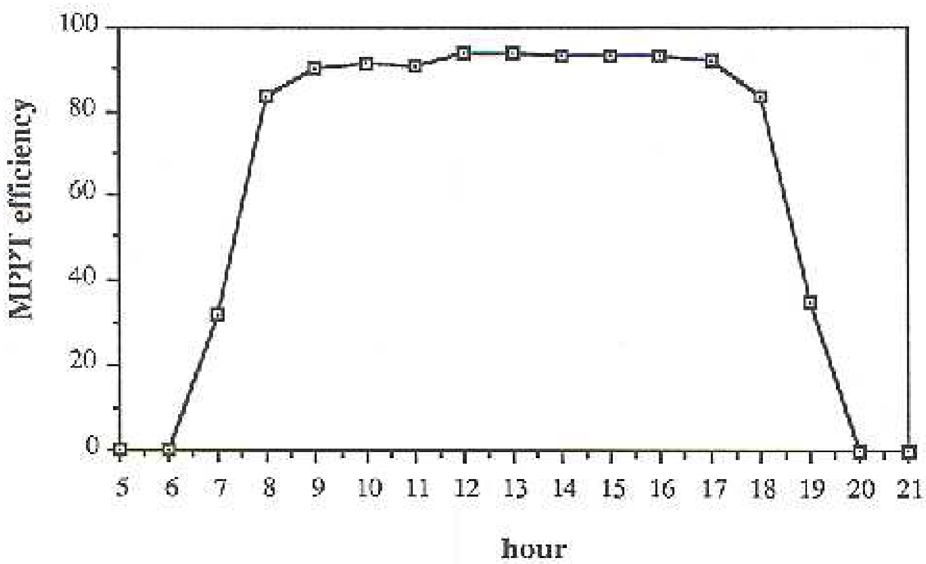

Figure 6.9 Maximum power point efficiency versus time of the real device

Chapter 7

CONCLUSIONS AND RECOMMENDATIONS

7.1 CONCLUSIONS

The Computer Models

The objective of this research is to develop detailed computer models which are capable of accurately simulating real PV system components accurately. The performance of the PV array model, the DC motor model and the hydraulic system model has been validated by a comparison with experimental data. The results of the simulation of the direct-coupled system's daily flow is within 2.5% of the real system. The maximum power point tracker model has been found to overestimate, by 9.7%, the performance of the compared PV system because of its ideally modelled behavior.

The developed models built a framework of the basic components of stand-alone PV systems including battery storage. The models are capable of simulating the performance over the entire operating range of the system including motor and pump starting conditions.

The modularity of *TRNSYS* facilitates parametric studies easily so that the performance of the individual components and their effects on the entire system can be examined in detail.

The PV Systems

Direct-coupled systems can be used for applications where the need for energy coincide with the daily appearance of solar energy. The studied motor-fan and motor-pump combinations showed that the separately excited motor is more compatible to the PV array than the series motor.

Maximum power tracking can lead to significant performance enhancements for direct-coupled systems, where the mismatch between load and PV array I-V curves is large. For well matched systems the effectiveness of a maximum power tracker is low or might even decrease the performance due to its non ideal behavior.

A battery acts like a voltage regulator when it is continuously connected to the system bus and can be used to operate the PV array close to the loci of maximum power points. Since the battery voltage varies little with state of charge and battery current, maximum power point tracking may not be very effective. The more sophisticated shunt type charge controller yields a better performance of the PV system than the simple series type charge controller.

7.2 RECOMMENDATIONS

As mentioned above, the basic models of PV stand alone systems have been developed. *TRNSYS* facilitates the implementation of user written components. Thus users can expand the developed models to any load model or other components, respecting the rules described in the previous Chapters.

Further work would include the development of PV system components modeling AC applications such as the operation with the utility grid, developing DC-AC converters, or other AC load components.

The temperature effect on battery performance was not considered. Including the temperature effect will improve the accuracy of battery modeling and should be subject of future research.

The ideal behavior of the MPPT model could be improved by adjusting the MPPT efficiency with the input-to-output voltage ratio of the MPPT instead of assuming a constant efficiency.

The array configuration, i.e. the modules wired in series and in parallel, is fixed throughout the simulation. The possibility of changing the configuration during a simulation may improve the PV system performance. The array series-parallel configuration could be adjusted in a way so that the locus of the maximum power point matches with the I-V curve of the load. This would also improve the control strategy of the series type charge controller. Trickle charge of the battery would be possible by partially shedding the array instead of disconnecting the entire array from the battery.

Appendix A

COMPONENTS

PV Array

```
C****************************************************************
   SUBROUTINE TYPE62(TIME,XIN,OUT,T,DTDT,PAR,INFO)
C     Version from: 11/16/89
C****************************************************************
C-----------------------------------------------------------------
C     This subroutine represents a four parameter model of
C     a Photovoltaic array. It is capable to predict the
C     complete current-voltage characteristic over the entire
C     operating voltage range of a flat-plate, non-sunconcen-
C     trated collector. While a series resistance is taken
C     into account, a shunt resistance is assumed to be infinite
C     and thus neglected in the model.
C     A routine is implemented which determines the voltage and
C     current at maximum power point.
C     The operating current is found for a given voltage,
C     irradiance and ambient temperature as input.
C     An option is provided to evaluate the series resistance with
C     the bisection method, if not given as input.
C     To overcome convergence problems which appear when simul-
C     ating direct coupled systems, a bisection method type of
C     convergence promotion is included. It is turned off, if
C     input FLAG is equal zero.
C****************************************************************
C**** Definition of the variables:
C**** TRNSYS specific variables:
C     XIN == input array
C     OUT == output array
C     PAR == parameters
C     TIME == simulation time
C     T,DTDT == not used in this component
C     S               == storage array
C     NSTORE == dimension of S
C     IAV == pointer within S
C     ISTORE == index
C     INFO == array to use TRNSYS internal information
C**** component specific variables:
```

```
C    A == completion factor
C    AREA == collector area [m^2]
C    CURRENT == function called current
C    DUMMY == auxiliary variable for convergence promotion
C    EFFREF == reference max. power efficiency
C    EG == bandgap energy [eV]
C    FF == fill factor
C    FIRST == auxiliary variable: allows that at first timestep
C          with solar radiation, V = VMP, otherwise the first
C          guess for V is V from the previous timestep
C    FLAG == switch: if FLAG=1 then convergence promotion is on
C             if FLAG=0 then convergence promotion is off
C    GAM == curve fit factor
C    I == current [amps]
C    IL == light current [A]
C    ILR == reference light current [A]
C    IMP == current at max. power [A]
C    IMR == reference current at max. power [A]
C    IO == reverse saturation current [A]
C    IOR == dito at reference [A]
C    ISC == short circuit current [A]
C    ISCR == dito at reference [A]
C    MEMO == memorizer if convergence promotion is on or off
C    MISC == temperature coefficient: short circuit current [A/K]
C    MVOC == temperature coefficient: open circuit voltage [V/K]
C    N == pointer to a relative address in program
C    NP == number of modules in parallel
C    NS == number of modules in series
C    P == power [W]
C    PMAX == power [W]
C    Q_BZ == electron charge/Boltzmann constant [C*K/J]
C    TA == ambient temperature [Kelvin (K)]
C    TANOCT == ambient temperature at NOCT [K]
C    TAU_AL == transmittance-absorptance product
C    TC == cell temperature [K]
C    TCR == reference cell temperature [K]
C    TCNOCT == cell temperature at NOCT [K]
C    SUN == irradiance [W/m^2]
C    SUNR == reference irradiance [W/m^2]
C    SUNNOCT == irradiance at NOCT [K]
C    V == voltage [volts]
C    VMP == max. power point voltage [V]
C    VMR == reference max. power point voltage [V]
C    VOC == open circuit voltage [V]
C    UTIL == utilization of the array: ratio of actual power to max. power
C**** variables used to determine the series resistance
C    ANEW,ALOW,AUP == A is the completion factor, the indexes
C              stands for the limits: lower and upper,  and for the current value: new
C    FNEW,FLOW,FUP == objective functions: at the interval
C              limits and at the current value
```

```
C    GAMNEW,GAMLOW,GAMUP == curve fit factor: at the interval
C                          limits and at the current value
C    IONEW,IOLOW,IOUP == saturation current: at the interval
C                          limits and at the current value
C    RS,RSNEW,RSLOW,RSUP == series resistance: at the interval
C                          limits and at the current value

C**** variables used in operating current calculation
C    F == objective function for Newton's method
C    FPRIME == first derivative of F
C    IOLD,INEW == iteration variables

C**** variables used in maximum power evaluation
C    F1 == objective function
C    F1P == first derivative of F1
C    IMXO,IMXN == iteration variables

C*********************************************************************
C**** declaration of the variables:
      IMPLICIT NONE
      REAL XIN,OUT,PAR,TIME,T,DTDT,S
      REAL I,CURRENT,V,IO,IOR,TA,
     &    A,Q_BZ,GAM,IL,ISC,TC,VOC,NS,NP,
     &    ISCR,VOCR,TCR,SUNR,VMR,IMR,MISC,EG,
     &    ILR,SUN,SERCELL,TCNOCT,TANOCT,SUNNOCT,
     &    AREA,TAU_AL,EFFREF,IMP,VMP,P,PMAX,MVOC,
     &    UTIL,VMI,VMA,VOLD,FF
      REAL ANEW,ALOW,AUP,FNEW,FLOW,FUP,GAMNEW,GAMLOW,GAMUP,
     &    IONEW,IOLOW,IOUP,RS,RSNEW,RSLOW,RSUP
      REAL IOLD,INEW,F,FPRIME
      REAL IMXO,IMXN,F1,F1P
        INTEGER INFO,FLAG,DUMMY,ISTORE,NSTORE,IAV,MEMO,N,FIRST
      DIMENSION XIN(4), OUT(10), PAR(18), INFO(10)
C*********************************************************************
      COMMON /STORE/ NSTORE,IAV,S(5000)
C**** store is used to store values from previous timestep
      COMMON /ARRAY/ GAM,TC,Q_BZ,IL,IO,RS,IMP
C**** array is used to transfer data to function 'current'
      COMMON /PARAM/ SERCELL,TCR,IMR,VMR,ISCR,VOCR,MVOC,MISC,EG
C**** param is used to transfer data to subroutine 'series'
C*********************************************************************
C---------------------------------------------------------------
C**** Set inputs
      SUN=XIN(1)/3.6
      TA=XIN(2)
      V=XIN(3)
      FLAG=XIN(4)
C---------------------------------------------------------------
C**** in this section a couple of checks on the inputs are performed. This is done at C**** the second and
```

```
following calls in timestep
   IF(INFO(7).GT.0)THEN
     DUMMY=INT(S(ISTORE+3))
     MEMO=INT(S(ISTORE+4))
     N=INT(S(ISTORE+5))
C****   all calculations are being scipped during time periods with no insolation
     IF(SUN.EQ.0.)THEN
      V=0.
      I=0.
C****      jump to the output section of the program
      GOTO 1000
     ELSE
C****      check on operation mode
      IF(FLAG.EQ.0)THEN
C**** convergence promotion is off if voltage is negative in system with battery C**** storage, cell is disconnected, i.e. output is zero current
       IF(V.LT.0.)THEN
        V=0.
        I=0.
       ELSE
C****         normal operation
C****         check if voltage greater than open circuit voltage
        VOC=OUT(8)
        IF(V.GT.VOC)THEN
         V=VOC
         I=0.
        ELSE
         I=CURRENT(V)
        ENDIF
       ENDIF
C****        jump to the output section of the program
       GOTO 1000
      ELSE
C****  this part is just invoked if a system with battery is operated and the battery C****  has to be disconnected temporarily to protect the battery from overcharging or C****  deep discharge. That means the system is changed to a direct coupled system. C****  Since this can happen after a few iterations running the battery system, the C****  convergence promoter, i.e. the number of iteration involved in the algorithm C****  has to be addressed relatively using N
       IF(MEMO.EQ.0)THEN
        N=INFO(7)
        MEMO=1
        V=VMP
        I=IMP
        GOTO 1000
       ENDIF
      ENDIF
     ENDIF
   ENDIF
C-----------------------------------------------------------------
C**** initial call in simulation
```

```
      IF (INFO(7).EQ.-1) THEN
       INFO(6)=10
       INFO(9)=0

C****   storage allocation: values from previous iteration are stored in the S-array
C****    S(ISTORE)  -- VOLD is voltage from previous iteration
C****    S(ISTORE+1) -- VMI  is lower voltage limit of convergence integral
C****    S(ISTORE+2) -- VMA  is upper limit on interval
C****    S(ISTORE+3) -- DUMMY
C****    S(ISTORE+4) -- MEMO
C****    S(ISTORE+5) -- N

       INFO(10)=6
       CALL TYPECK(1,INFO,4,18,0)
       ISTORE=INFO(10)
       S(ISTORE)=0.
       S(ISTORE+1)=0.
       S(ISTORE+2)=0.
       S(ISTORE+3)=0.
       S(ISTORE+4)=0.
       S(ISTORE+5)=0.
        Q_BZ=11604.45
C****   set parameters   ***************************
       ISCR=PAR(1)
       VOCR=PAR(2)
       TCR=PAR(3)
       SUNR=PAR(4)
       VMR=PAR(5)
       IMR=PAR(6)
       MISC=PAR(7)
       MVOC=PAR(8)
       SERCELL=PAR(9)
       NS=PAR(10)
       NP=PAR(11)
       TCNOCT=PAR(12)
       TANOCT=PAR(13)
       SUNNOCT=PAR(14)
       AREA=PAR(15)
       TAU_AL=PAR(16)
       EG=PAR(17)
       RS=PAR(18)

       IF (RS.LT.0) THEN
C****    in this case RS is not provided as a parameter and
C****    has to be evaluated. This is done in subroutine "series"
         CALL SERIES(RS,Q_BZ)
       ENDIF

C****   evaluation of the 3 unknowns at reference condition
       GAM=Q_BZ*(VMR-VOCR+IMR*RS)/(TCR*LOG(1.-IMR/ISCR))
```

```
      ILR=ISCR
      IOR=ILR/EXP(Q_BZ*VOCR/(GAM*TCR))

C****   set up parameters for the entire array
      ILR=NP*ILR
      IOR=NP*IOR
      GAM=NS*GAM
      RS=(NS/NP)*RS

      A=GAM/(NS*SERCELL)
      MEMO=0
      FIRST=0
      GOTO 1000
C--------------------------------------------------------------
C**** first call in timestep
     ELSE IF (INFO(7).EQ.0) THEN
        VOLD=S(ISTORE)
C****   evaluation of cell temperature from NOCT conditions
      EFFREF=IMR*VMR/(SUNR*AREA)
      TC=TA+(SUN*(TCNOCT-TANOCT)/SUNNOCT*(1.-EFFREF/TAU_AL))

C****   this part calculates how IL and IO vary with temp. and SUN
      IL=(SUN/SUNR)*(ILR+MISC*NP*(TC-TCR))
      IF(IL.LT.0.0) IL=0.0
      IO=IOR*((TC/TCR)**3)*EXP((Q_BZ*EG/(A))*((1./TCR)-(1./TC)))

C****   Open circuit voltage
      VOC=GAM*TC/Q_BZ*LOG(IL/IO+1.)
C****   Short circuit current
      ISC=IL

C****   this part calculates the max-power current & voltage using NEWTON's
C****   method to solve for the max-power voltage and max-power current
      IMXO=0.0
C****   first guess for max. power point current
      IMXN=SUN/SUNR*NP*(IMR+MISC*(TC-TCR))
      DO WHILE (ABS(IMXN-IMXO).GT.0.001)
       IMXO=IMXN
       F1=IMXO+(IMXO-IL-IO)*(LOG((IL-IMXO+IO)/IO)-IMXO*Q_BZ*RS/
    >   (GAM*TC))/(1.+(IL-IMXO+IO)*(Q_BZ*RS/(GAM*TC)))
       F1P=2.+(LOG((IL-IMXO+IO)/IO)-(Q_BZ*RS*IMXO/(GAM*TC)))/
    >   ((1.+(IL-IMXO+IO)*(Q_BZ*RS/(GAM*TC)))**2.)
       IMXN=IMXO-(F1/F1P)
      END DO
      IMP=IMXN
      VMP=LOG(1.+(IL-IMP)/IO)*(TC*GAM/Q_BZ)-IMP*RS
      IF ((VOC.GT.0.).AND.(ISC.GT.0)) THEN
C****     Fill factor
        FF=VMP*IMP/VOC/ISC
      ELSE
```

```
      FF=0.
     ENDIF
C*****   all calculations are being scipped during time periods
C****    with no insolation
      IF(SUN.EQ.0.)THEN
       V=0.
       I=0.
      ELSE
C****     check on operation mode
       IF(FLAG.EQ.0)THEN
C****       convergence promotion is off
C****       if voltage is negative in system with battery storage,
C****       cell is disconnected, i.e. output is zero current
        IF(V.LT.0.)THEN
         V=0.
         I=0.
        ELSE
C****        normal operation
C****        check if voltage greater than open circuit voltage
         IF(V.GT.VOC)THEN
          V=VOC
          I=0.
         ELSE
          I=CURRENT(V)
         ENDIF
        ENDIF

       ELSE
C**** convergence promotion is on, i.e. direct coupled mode
        MEMO=1
        N=INFO(7)
        IF(FIRST.EQ.0)THEN
C**** at the first timestep of the simulation at which solar radiation occurs, the first C**** guess on V is V
at max. power point
         V=VMP
         I=IMP
         FIRST=1
        ELSE
C**** the first guess foe each timestep on V is the voltage at operating point from the C**** previous
timestep
         V=VOLD
         I=CURRENT(V)
        ENDIF
       ENDIF
      ENDIF
     PMAX=IMP*VMP
      DUMMY=0
      GOTO 1000
C-----------------------------------------------------------------
C**** second call in timestep, first iterative call
```

```
      ELSE IF (INFO(7).EQ.(N+1)) THEN

C****   here begins the convergence promotion algorithm
C****   the first guess was the voltage from the previous
C****   timestep. The voltages for the next two iterations
C****   are tracked before the promoter intervenes. Until
C****   then a protection prevents the voltage to jump
C****   out of bounds, i.e. if the input voltage is
C****   greater than open circuit voltage it is set equal
C****   to it. The convergence promotion algorithm is
C****   basically the same as the bisection method, it
C****   cuts an interval in half until a sufficient tolerance
C****   is obtained.

        VOLD=S(ISTORE)
        VOC=OUT(8)
        IF(V.GT.VOLD)THEN
         IF(V.GT.VOC)THEN
          DUMMY=4
          VMI=VOLD
          VMA=VOC
          V=(VMI+VMA)/2.
         ELSE
          DUMMY=1
          VMI=VOLD
         ENDIF
        ELSE
         DUMMY=2
         VMA=VOLD
        ENDIF
        I=CURRENT(V)
        GOTO 1000
C------------------------------------------------------------------
C**** third call in timestep
      ELSE IF(INFO(7).EQ.(N+2))THEN
        VOLD=S(ISTORE)
        VOC=OUT(8)
        IF(DUMMY.EQ.1)THEN
         IF(V.GT.VOLD)THEN
          DUMMY=3
         ELSE
          VMA=VOLD
          IF(V.GT.VMI)THEN
           VMI=V
          ENDIF
          V=(VMI+VMA)/2.
         ENDIF
        ELSEIF(DUMMY.EQ.4)THEN
         IF(V.GT.VOLD)THEN
          VMI=VOLD
```

```
       IF(V.GT.VOC)THEN
        VMA=VOC
       ELSE
        VMA=V
       ENDIF
       V=(VMI+VMA)/2.
      ELSE
       VMA=VOLD
       IF(V.GT.VMI)THEN
        VMI=V
       ENDIF
       V=(VMI+VMA)/2.
      ENDIF
     ELSE
     IF(V.GT.VOLD)THEN
      VMI=VOLD
      IF(V.LT.VMA)THEN
       VMA=V
      ENDIF
      V=(VMI+VMA)/2.
     ELSE
      DUMMY=3
     ENDIF
     ENDIF
     I=CURRENT(V)
     GOTO 1000
C-----------------------------------------------------------------
C**** fourth and following calls in timestep
    ELSE
     IF(DUMMY.NE.3)THEN
     VMI=S(ISTORE+1)
      VMA=S(ISTORE+2)
      VOLD=S(ISTORE)
      VOC=OUT(8)
      IF(V.GT.VOLD)THEN
       VMI=VOLD
       IF(V.LT.VMA)THEN
        VMA=V
       ENDIF
     ELSE
      VMA=VOLD
      IF(V.GT.VMI)THEN
       VMI=V
      ENDIF
     ENDIF
      V=(VMI+VMA)/2.
    ENDIF
C****   in this case iteration is done with TRNSYS
   ENDIF
```

```
C--------------------------------------------------------------------
    I=CURRENT(V)
1000 CONTINUE
    P=I*V
C**** updating and storing voltage values for the
C**** convergence promoter
C**** voltage is only stored at timesteps with incident radiation
    IF(SUN.NE.0.)THEN
     VOLD=V
    ENDIF
    S(ISTORE)=VOLD
    S(ISTORE+1)=VMI
    S(ISTORE+2)=VMA
    S(ISTORE+3)=DUMMY
    S(ISTORE+4)=MEMO
    S(ISTORE+5)=N

C**** SET OUTPUTS  ************************************
    OUT(1)=V
    OUT(2)=I
    OUT(3)=P
    OUT(4)=PMAX
    IF(PMAX.NE.0.)THEN
     UTIL=P/PMAX
    ELSE
     UTIL=0.
    ENDIF
    OUT(5)=UTIL
    OUT(6)=VMP
    OUT(7)=IMP
    OUT(8)=VOC
    OUT(9)=ISC
    OUT(10)=FF
   END

C*******************************************************************
    FUNCTION CURRENT(V)
C**** function computes the current for a given voltage
C**** the implicit equation is solved using Newton's
C**** method
    IMPLICIT NONE
    COMMON /ARRAY/ GAM,TC,Q_BZ,IL,IO,RS,IMP
    REAL IOLD,INEW,F,FPRIME,IMP,I,CURRENT,V
    REAL GAM,TC,Q_BZ,IL,IO,RS

    IOLD=0.0
C**** first guess is max.power point current
    INEW=IMP
    DO WHILE (ABS(INEW-IOLD).GT.1.E-3)
     IOLD=INEW
```

```
    F=IL-IOLD-IO*(EXP(Q_BZ*(V+IOLD*RS)/(GAM*TC))-1.)
    FPRIME=-1.-IO*RS*Q_BZ/GAM/TC*EXP(Q_BZ*(V+IOLD*RS)
   >       /GAM/TC)
    INEW=IOLD-(F/FPRIME)
   END DO
   I=INEW
   CURRENT=I
   END

C****************************************************************
   SUBROUTINE SERIES(RS,Q_BZ)
C**** determination of series resistance using bisection method

   IMPLICIT NONE
   COMMON /PARAM/ SERCELL,TCR,IMR,VMR,ISCR,VOCR,MVOC,MISC,EG

   REAL SERCELL,TCR,IMR,VMR,ISCR,VOCR,MVOC,MISC,EG,Q_BZ
   REAL ANEW,ALOW,AUP,FNEW,FLOW,FUP,GAMNEW,GAMLOW,GAMUP
   REAL IONEW,IOLOW,IOUP,RS,RSNEW,RSLOW,RSUP

C**** parameters at the upper limit of the convergence interval
   RSUP=((SERCELL*TCR*LOG(1.-IMR/ISCR)/Q_BZ)+VOCR-VMR)/IMR
   AUP=1.
   GAMUP=SERCELL
   IOUP=ISCR*EXP(-Q_BZ*VOCR/(GAMUP*TCR))
C**** parameters at the lower limit of the convergence interval
   RSLOW=0.0
   GAMLOW=Q_BZ*(VMR-VOCR)/(TCR*LOG(1.-IMR/ISCR))
   ALOW=GAMLOW/SERCELL
   IOLOW=ISCR*EXP(-Q_BZ*VOCR/(GAMLOW*TCR))
   DO WHILE ((ABS(RSUP-RSLOW)).GT.0.0005)
    RSNEW=0.5*(RSUP+RSLOW)
    GAMNEW=Q_BZ*(VMR-VOCR+IMR*RSNEW)/(TCR*LOG(1.-IMR/ISCR))
    ANEW=GAMNEW/SERCELL
    IONEW=ISCR*EXP(-Q_BZ*VOCR/(GAMNEW*TCR))
    FUP=-MVOC+(GAMUP/Q_BZ)*(LOG(1.+ISCR/IOUP)+(TCR/(ISCR+
   > IOUP))*(MISC-ISCR*((Q_BZ*EG/(AUP*TCR*TCR))+3./TCR)))
    FLOW=-MVOC+(GAMLOW/Q_BZ)*(LOG(1.+ISCR/IOLOW)+(TCR/(ISCR+
   > IOLOW))*(MISC-ISCR*((Q_BZ*EG/(ALOW*TCR*TCR))+3./TCR)))
    FNEW=-MVOC+(GAMNEW/Q_BZ)*(LOG(1.+ISCR/IONEW)+(TCR/(ISCR+
   > IONEW))*(MISC-ISCR*((Q_BZ*EG/(ANEW*TCR*TCR))+3./TCR)))
    IF((FLOW*FNEW).LT.0.0) RSUP=RSNEW
    IF((FLOW*FNEW).GT.0.0) RSLOW=RSNEW
    GAMUP=Q_BZ*(VMR-VOCR+IMR*RSUP)/(TCR*LOG(1.-IMR/ISCR))
    AUP=GAMUP/SERCELL
    IOUP=ISCR*EXP(-Q_BZ*VOCR/(GAMUP*TCR))
    GAMLOW=Q_BZ*(VMR-VOCR+IMR*RSLOW)/(TCR*LOG(1.-IMR/ISCR))
    ALOW=GAMLOW/SERCELL
    IOLOW=ISCR*EXP(-Q_BZ*VOCR/(GAMLOW*TCR))
   END DO
```

```
RS=RSNEW
END
```

Maximum Power Point Tracker

```
C*****************************************************************
      SUBROUTINE TYPE65(TIME,XIN,OUT,T,DTDT,PAR,INFO)
C     version from: 01/12/89
C*****************************************************************
C**** Subroutine represents a maximum power tracking device
C**** for a PV power system
C**** the maximum power tracker developed in this component is
C**** basically a DC - DC converter. The goal is to find the
C**** optimal conversion ratio to match the converted max.power
C**** voltage to the load voltage.  A convergence promoter asures
C**** convergence an accelerates the iterative process for
C**** direct coupled systems: Mode = 1. Convergence promotion is
C**** done on the output variable Ilead.
C**** The convergence promoter is basically a bisection method
C*****************************************************************

C**** variables:
C    ICELL -- current from PV array [amps]
C    ILOAD --  current to load [amps]
C    VLOAD -- load voltage [V]
C    RATIO -- ratio of cell voltage to motor voltage
C          or ratio of cell current to motor current
C    X -- variable to store current value for
C         convergence promoter method
C    XNEW -- new value >> convergence promoter
C    XPOS -- limit >> convergence promoter
C    XNEG -- limit >> convergence promoter
C    DUMMY -- variable used as a control function to obtain
C          values from the two previous timesteps
C    EFF -- efficiency of maximum power tracker
C    MODE -- determines operating mode either direct coupled
C         or not
C    G -- variable used in the con. promoter
C    GUARD -- control variable
C*****************************************************************

      IMPLICIT NONE
      INTEGER INFO
      REAL TIME,XIN,OUT,T,DTDT,PAR
      REAL ICELL,ILOAD,VCELL,VLOAD,X,XNEW,XPOS,XNEG,RATIO
      REAL G,EFF,MODE,DUMMY,GUARD
      DIMENSION XIN(4), OUT(6),INFO(10),PAR(1)
C*****************************************************************
      INFO(6)=7
      INFO(9)=0
      CALL TYPECK(1,INFO,4,1,0)

C**** Set inputs
```

```
      ICELL=XIN(1)
      VCELL=XIN(2)
      VLOAD=XIN(3)
      MODE=XIN(4)

C---------------------------------------------------------
C**** Initial call of component
      IF(INFO(7).LT.0)THEN
C****  SET PARAMETERS
        EFF=PAR(1)

C****   Initialization of some variables
        X=0.
        XPOS=0.
        XNEG=0.
        RATIO=0.
        DUMMY=0.
        GUARD=0.
C---------------------------------------------------------
C**** first and following calls in time step
      ELSE
        GUARD=OUT(7)
        DUMMY=OUT(3)
        IF(VCELL.EQ.0.)THEN
C****      scip all calculations
          RATIO=0.
          ILOAD=0.
        ELSE

C****      First call in time step
          IF(INFO(7).EQ.0)THEN
            GUARD=MODE
C****        dummy addresses the convergence promoting algorithm
          DUMMY=0.
        ENDIF

C****      guard observes the mode input
C****      if mode changes from 0 to 1, indicating
C****      a switch from battery system mode to direct coupled
C****      system mode, then convergence promoter starts over
          IF(GUARD.NE.MODE)THEN
            DUMMY=0.
            GUARD=MODE
          ENDIF
          IF(MODE.EQ.0.)THEN
           IF(VLOAD.GT.0.)THEN
           RATIO=VCELL/VLOAD
           ILOAD=ICELL*RATIO*EFF
           ELSE
             RATIO=0.
```

```
        ILOAD=0.
       ENDIF
       ELSE
C****        get values from previous iteration
         X=OUT(4)
         XPOS=OUT(5)
         XNEG=OUT(6)

         IF(VLOAD.EQ.0.)THEN
C****         prevents devision by zero, see below
          VLOAD=VCELL
         ENDIF

       RATIO=VCELL/VLOAD
       ILOAD=ICELL*RATIO*EFF

C**** here begins the convergence promoter
        IF(DUMMY.EQ.0.)THEN
          XNEW=ILOAD
          DUMMY=1.
         ELSEIF(DUMMY.EQ.1.)THEN
          XNEW=ILOAD
          G=XNEW-X
           IF(G.GT.0.)THEN
            XPOS=XNEW
            XNEG=X
          ELSE
             XPOS=X
             XNEG=XNEW
          ENDIF
          XNEW=(XPOS+XNEG)/2.
          DUMMY=2.

        ELSE

          XNEW=ILOAD
          G=XNEW-X
           IF(G.GT.0.)THEN
            XNEG=X
          ELSE
             XPOS=X
          ENDIF
          XNEW=(XPOS+XNEG)/2.
         ENDIF
        ILOAD=XNEW
        X=XNEW
       ENDIF
      ENDIF
     ENDIF
C------------------------------------------------------------------
```

```
C**** SET OUTPUTS
    OUT(1)=ILOAD
    OUT(2)=RATIO
C**** following outputs serve as storage
    OUT(3)=DUMMY
    OUT(4)=X
    OUT(5)=XPOS
    OUT(6)=XNEG
    OUT(7)=GUARD
    END
```

DC Motors

```
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C    SUBROUTINE TYPE66(TIME,XIN,OUT,T,DTDT,PAR,INFO)
C     Version from: 12/17/89
C-----------------------------------------------------------
C    This subroutine represents a model a for a separately
C     exited DC motor (permanent magnet or any other const.
C    current field source) and a series DC motor. It is
C     an analytical model based on fundamental motor equations.
C    For a direct coupled PV system MODE should be set equal to
C     one and current and speed will be the significant inputs.
C    For a system with battery storage MODE should be set equal
C      to zero, i.e. voltage and speed will be significant inputs.
C     To support convergence between the motor component and
C     any motor load component a convergence promotor is included.
C****************************************************************
C****************************************************************
C    Variables:
C    CSTAT -- static friction [Nm]
C    CVISC -- viscous friction [Nm*sec]
C    CHANGE -- auxiliary variables
C    FAIL -- to output error message
C           =1: max. speed is exceeded
C           =2: rated current is exceeded
C           =3: max. current is exceeded
C    ETOL,TOL -- tolerances used in the convergence promoter
C     EFLAG -- variable to keep track of significant inputs
C     FLAG -- determines whether bisection or secant method
C           is being continued
C    G -- objective function
C    GLAST,GNEG,GPOS -- objective functions at the limits
C                and from the previous timestep
C    I -- Current through motor [amps]
C    IMAX -- max. current [amps]
C    IRAT -- rated current [amps]
C     KF -- motor constant,relates the permanent magnet flux
```

```
C         through the armature current and the emf [Nm/amp]
C    MAF -- mutual inductance [henry]
C    MEMO -- keeps track of the operating mode
C    MODE -- determines whether I or V and respectively N or T
C           are significant inputs
C              =0: V and N are significant
C              =1: I and N are significant
C    N  -- motor speed [1/sec]
C    NMAX -- max. speed [1/sec]
C    NN -- relative addresse used in convergence promoter
C    RA -- motor armature resistance [ohms]
C    RF -- field resistance [ohms]
C    TEM -- motor torque (electromagnetical) [Nm]
C    TLOSS -- loss torque
C    TSHAFT -- shaft torque
C    TYPE -- determines type of DC motor
C              =1: separately excited
C              =2: series
C    V  -- terminal voltage [volts]
C    X,XLAST,XNEG,XPOS,XNEW -- converging variable - torque
C*****************************************************************

      IMPLICIT NONE
      INTEGER INFO,TYPE,FAIL
      INTEGER NSTORE,IAV,ISTORE
      INTEGER NN,MEMO,FLAG
      REAL TIME,XIN,OUT,T,DTDT,PAR,S
      REAL V,I,N,TEM,RA,KF,PI,IRAT,TLOSS,TSHAFT
      REAL MAF,RF,IMAX,NMAX,CSTAT,CVISC,MODE
      REAL PEL,PMECH,EFF,TOL,ETOL,CHANGE,EFLAG
      REAL GLAST,GNEG,GPOS,XLAST,X,XNEW,XPOS,XNEG,G
      DIMENSION XIN(5), OUT(8), PAR(9), INFO(10)

      COMMON /STORE/ NSTORE,IAV,S(5000)
C*****************************************************************
C-----------------------------------------------------------------
C**** initial call in simulation
      IF(INFO(7).LT.0)THEN
C****   accurate expression for pi
       PI=ACOS(-1.)

       INFO(6)=8
       INFO(9)=0

C****  SET PARAMETERS
       TYPE=PAR(1)
       IF(TYPE.EQ.1)THEN
        CALL TYPECK(1,INFO,5,8,0)
          ELSE
        CALL TYPECK(1,INFO,5,9,0)
```

```
         ENDIF
NMAX=PAR(2)
         IRAT=PAR(3)
         IMAX=PAR(4)
IF(TYPE.EQ.1)THEN
     KF=PAR(5)
         ELSE
          MAF=PAR(5)
         ENDIF
         CSTAT=PAR(6)
         CVISC=PAR(7)
    RA=PAR(8)
         IF(TYPE.GT.1)THEN
          RF=PAR(9)
         ENDIF
         INFO(10)=11
         IF(TYPE.GT.1)THEN
          CALL TYPECK(1,INFO,5,9,0)
         ELSE
          CALL TYPECK(1,INFO,5,8,0)
         ENDIF
         ISTORE=INFO(10)

C****   initialization of storage values
         S(ISTORE)=0.
         S(ISTORE+1)=0.
         S(ISTORE+2)=0.
         S(ISTORE+3)=0.
         S(ISTORE+4)=0.
         S(ISTORE+5)=0.
         S(ISTORE+6)=0.
         S(ISTORE+7)=0.
         S(ISTORE+8)=0.
         S(ISTORE+9)=0.
         S(ISTORE+10)=0.

         TOL=0.001
         ETOL=0.01
         NN=0
         MEMO=0
C--------------------------------------------------------
C**** first and following calls in timestep
   ELSE
         NN=S(ISTORE+7)
         MEMO=S(ISTORE+8)
         FLAG=S(ISTORE+9)
         EFLAG=S(ISTORE+10)
    FAIL=0

C****   set inputs
```

```
      I=XIN(1)
      V=XIN(2)
      N=XIN(3)
      TSHAFT=XIN(4)
      MODE=XIN(5)

      IF(INFO(7).EQ.0)THEN
       MEMO=INT(MODE)
       FLAG=0
       NN=0
       IF(MODE.EQ.1.)THEN
         EFLAG=I
       ELSE
         EFLAG=V
       ENDIF
      ENDIF

C****   if mode changes during timestep, the convergence
C****   promoter is reset to the first step
      IF(MEMO.NE.INT(MODE))THEN
       NN=INFO(7)
       MEMO=INT(MODE)
      ENDIF

C****   if significant inputs change during timestep, the convergence
C****   promoter is reset to the first step
      IF((MODE.EQ.1.).AND.(I.NE.EFLAG))THEN
       NN=INFO(7)
       EFLAG=I
      ELSEIF((MODE.EQ.0.).AND.(V.NE.EFLAG))THEN
       NN=INFO(7)
       EFLAG=V
      ENDIF

C-----------------------------------------------------------
C****   characteristic motor equations
      TLOSS=CSTAT+CVISC*N
      IF(MODE.EQ.1) THEN
C****   current and speed serve as input, voltage and
C****   torque are computed
       IF(TYPE.EQ.1)THEN
C****     separately excited motor
      V=KF*2.*PI*N+I*RA
      TEM=KF*I
       ELSEIF(TYPE.EQ.2)THEN
C****     series motor
        V=I*(2.*PI*N*MAF+RA+RF)
        TEM=MAF*I**2
       ENDIF
```

```
      ELSE
C****     speed and voltage serve as input, torque and current
C****     are computed.
          IF(TYPE.EQ.1)THEN
C****      separately excited motor
       I=(V-KF*2.*PI*N)/RA
      TEM=KF*I
          ELSEIF(TYPE.EQ.2)THEN
C****      series motor
           I=V/(2.*PI*N*MAF+RA+RF)
           TEM=MAF*I**2
          ENDIF
      ENDIF
      TSHAFT=TEM-TLOSS
C****   check on TSHAFT
      IF(TSHAFT.LT.0.)THEN
       TSHAFT=0.
      ENDIF
C------------------------------------------------------------
C****   Convergence promotion on torque
         X=S(ISTORE)
         GPOS=S(ISTORE+1)
         XPOS=S(ISTORE+2)
         GNEG=S(ISTORE+3)
         XNEG=S(ISTORE+4)
         GLAST=S(ISTORE+5)
         XLAST=S(ISTORE+6)

         IF(INFO(7).EQ.NN)THEN
C****      successive substitution
          XNEW=TSHAFT
        FLAG=0
         ELSEIF((INFO(7).EQ.NN+1).OR.(INFO(7).EQ.NN+2))THEN
C****      successive substitution
          XNEW=TSHAFT
          G=XNEW-X
          IF(G.GT.0.)THEN
           GPOS=G
           XPOS=X
          ELSE
           GNEG=G
           XNEG=X
          ENDIF
         ELSE
          XNEW=TSHAFT
          G=XNEW-X

C****     Convergence check
       IF(ABS(G).GT.TOL)THEN
           IF((ABS(ABS(G)-ABS(GNEG)).LT.ETOL).OR.(ABS(ABS(G)
```

```
    &      -ABS(GPOS)).LT.ETOL).OR.(FLAG.EQ.1))THEN
C****        use bisection method
             IF(G.GT.0.)THEN
              XPOS=X
             ELSE
              XNEG=X
             ENDIF

             XNEW=(XPOS+XNEG)/2.
             FLAG=1
            ELSE
C****        use secant method unless residuals increase
             IF(INFO(7).EQ.NN+3)THEN
C****          two most reasonable previous guesses are used
              IF((G.GT.GNEG).AND.(G.GT.GPOS))THEN
               G=GPOS
               X=XPOS
               GLAST=GNEG
               XLAST=XNEG
              ELSE
               IF(G.LT.0.)THEN
                IF(ABS(G).GT.ABS(GNEG))THEN
                         GLAST=GNEG
                         XLAST=XNEG
                        ENDIF
               ELSE
                IF(ABS(G).GT.GPOS)THEN
                         GLAST=GPOS
                         XLAST=XPOS
                ENDIF
               ENDIF
              ENDIF
             ELSE
                   IF(G.GT.GLAST)THEN
C****                switch to bisection method
               XNEW=(XPOS+XNEG)/2.
                     FLAG=1
              ENDIF
             ENDIF
             CHANGE=G*(X-XLAST)/(G-GLAST)
             XNEW=X-CHANGE

         IF((XNEW.GT.XNEG).OR.(XNEW.LT.XPOS))THEN
C****          switch to bisection
               XNEW=(XPOS+XNEG)/2.
          FLAG=1
         ENDIF
            ENDIF
        TSHAFT=XNEW
       ELSE
```

```
C****      solution is found
        TSHAFT=X
       ENDIF
          ENDIF

          XLAST=X
          GLAST=G
          X=XNEW
          S(ISTORE)=X
          S(ISTORE+1)=GPOS
          S(ISTORE+2)=XPOS
          S(ISTORE+3)=GNEG
          S(ISTORE+4)=XNEG
          S(ISTORE+5)=GLAST
          S(ISTORE+6)=XLAST
          S(ISTORE+7)=NN
          S(ISTORE+8)=MEMO
          S(ISTORE+9)=FLAG
          S(ISTORE+10)=EFLAG
   ENDIF

   IF(N.GT.NMAX)THEN
          FAIL=1
   ENDIF
   IF(I.GT.IRAT)THEN
          IF(I.GT.IMAX)THEN
           FAIL=3
          ELSE
           FAIL=2
          ENDIF
   ENDIF

   PEL=I*V
   PMECH=2.*PI*N*TSHAFT
   IF(PEL.NE.0.)THEN
          EFF=PMECH/PEL
   ELSE
          EFF=0.
   ENDIF

C**** SET OUTPUTS
   OUT(1)=I
   OUT(2)=V
   OUT(3)=N
   OUT(4)=TSHAFT
   OUT(5)=PEL
   OUT(6)=PMECH
   OUT(7)=EFF
   OUT(8)=FAIL
```

```
      END
```

Centrifugal Fan

```
C********************************************************************
      SUBROUTINE TYPE67(TIME,XIN,OUT,T,DTDT,PAR,INFO)
C     Version from: 12/13/89
C--------------------------------------------------------------------
C**** This Type represents a Ventilator type load, which is
C     designed to run with an electric motor.
C********************************************************************
C**** Nomenclature:
C**** TRNSYS specific variables:
C     XIN == input array
C     OUT == output array
C     PAR == parameters
C     TIME == simulation time
C     T,DTDT == not used in this component
C     S    == storage array
C     NSTORE == dimension of S
C     IAV == pointer within S
C     ISTORE == index
C     INFO == array to use TRNSYS internal information
C**** C1,C2,C3 -- constants
C     COUNTER -- counting the number of iterations
C     CHANGE -- auxilary variable
C     EPS -- error tolerance for newton iteration
C     FAIL -- message if operating conditions are out of range
C             :0 = everything is okay
C             :1 = max. speed is exceeded
C             :2 = Newton iteration did not converge within specified tolerance
C     FN -- objective function for Newton iteration
C     FPRIMEN -- first derivative of the objective function
C     MAXIT -- the limit on the number of iterations to find the speed
C     MODE -- determines whether torque or speed is significant
C             :1 = torque is significant
C             :2 = speed is significant
C     N -- Speed
C     NINIT -- initial value of a speed for the iterative process
C             to find the motor speed for a given torque
C     NMAX -- maximum permitted speed
C     NOLD -- speed from previous iteration
C     P -- Fan power [watts]
C     PI - pi
C     TL -- Torque
C********************************************************************
      IMPLICIT NONE
      INTEGER ISTORE,NSTORE,IAV
```

```
      INTEGER INFO,COUNTER,MAXIT,FAIL
      REAL TIME,XIN,OUT,T,DTDT,PAR,S
      REAL TL,N,NINIT,C1,C2,C3,FN,FPRIMEN,NNEW,CHANGE
      REAL NMAX,EPS,PI,NOLD,P
      DIMENSION XIN(1), OUT(4), PAR(4), INFO(10)
C***************************************************************
      COMMON /STORE/ NSTORE,IAV,S(5000)
C---------------------------------------------------------------
C**** Initial call in simulation
      IF (INFO(7).LT.0) THEN

       INFO(6)=4
       INFO(9)=0

C****   storage allocation: speed from previous iteration
C****   is stored in the S-array
C****  S(ISTORE) -- NOLD
       INFO(10)=1
       CALL TYPECK(1,INFO,1,4,0)
       ISTORE=INFO(10)

C****   set parameters
       C1=PAR(1)
       C2=PAR(2)
       C3=PAR(3)
       NMAX=PAR(4)

          NINIT=0.01*NMAX
          NOLD=NINIT
       MAXIT=30
          EPS=0.0001

C****   accurate expression for pi
       PI=ACOS(-1.)
C---------------------------------------------------------------
C**** first and following calls in timestep
      ELSE
       FAIL=0
C****   set inputs
       TL=XIN(1)

       IF(TL.LT.C1)THEN
        N=0.
       ELSE

C****     Iteration using Newton's method to find speed
        COUNTER=0
           NINIT=S(ISTORE)
        NNEW=NINIT
        N=NINIT
```

```
100    CONTINUE
       FN=C1+C2*((2.*PI*N)**C3)-TL
        FPRIMEN=2.*PI*C3*C2*((2.*PI*N)**(C3-1.))
      CHANGE=-FN/FPRIMEN
      IF (ABS(CHANGE).GE.EPS) THEN
       NNEW=N+CHANGE
       IF (COUNTER.GE.MAXIT) THEN
         FAIL=2
       ELSE
        N=NNEW
        COUNTER=COUNTER+1
        GOTO 100
       ENDIF
      ELSE
       N=NNEW
      ENDIF
     ENDIF
    ENDIF
C-------------------------------------------------------------

C**** Control on speed, speed can't be negative
    IF (N.LT.0) THEN
     N=0.
    ELSE
C****    control on max.speed
     IF(N.GT.NMAX)THEN
      FAIL=1
     ENDIF
    ENDIF

    IF(N.NE.0.)THEN
        NOLD=N
    ENDIF

    S(ISTORE)=NOLD
    P=2.*PI*N*TL

C**** set outputs
    OUT(1)=N
    OUT(2)=TL
    OUT(3)=P
    OUT(4)=FAIL
    END
```

Centrifugal Pump

```
C*********************************************************************
    SUBROUTINE TYPE68(TIME,XIN,OUT,T,DTDT,PAR,INFO)
```

```
C    Version from: 12/14/89
C---------------------------------------------------------------
C**** This Type represents a centrifugal pump, which is designed
C    to run with an electric motor.
C    The model requires two sets of data: head-flowrate data at
C    reference speed and efficiency-flowrate data at reference
C    speed. A linear regression is performed to fit a curve
C    through those data. The normal equations are solved using
C    Gaussian elimination with partial pivoting. Newtons method
C    is implemented to solve a system of nonlinear equations.
C*****************************************************************
C**** Variables:
C   TL -- Torque [Nm]
C   N  -- Speed [1/s]
C   Q  -- Volumetric flowrate [m^3/s]
C   H  -- total dynamic head [m]
C   ETA -- pump efficiency
C   TOFF,QOFF,NOFF,ETAOFF -- T,Q,N,ETA at shutoff condition
C   TREQ -- required torque to provide flow
C   NREF -- Reference speed
C   NINIT -- initial speed for Newton's method
C   NMAX -- maximum speed of pump
C   QINIT -- initial flowrate for Newton's method
C   QNOM -- nominal design flowrate [m^3/hour]
C   MAXIT -- maximum number of iterations to find the
C          speed and flowrate in Newton's method
C   COUNTER -- counting the number of iterations
C   POFF -- Shaftpower at shutoff condition at reference speed POFF is input as
C   parameter. If no information is available, Poff should be set equal some negative
C   number
C   EPS -- error tolerance in Newton's method
C   A1-H1 -- coefficients for the H=f(Q,N) and ETA=f(Q,N) relations
C   S1-S8 -- defined constants in the equations
C   RHOG -- density, rho, times acceleration const., g
C   CONST -- specific weight of water(rho*g) devided by two times pi
C   X -- two dimensional array contains N and Q -- old values
C      X1=N  X2=Q
C   XNEW -- same as X, but containing new values
C   F -- two dimensional array contains values of the objectiv function
C   J -- Jacobian matrix (2*2) contains the derivatives
C   JINV -- inverse Jacobian matrix
C   DET -- determinant of the Jacobian matrix
C   LOOP -- control variable
C   PHYD -- water power
C   PMECH -- mechanical (input) power
C   FAIL -- error message
C          :0 - everything is okay
C          :1 - maximum pump speed is exceeded
C          :2 - newton method could not find proper solution
C          :3 - newton method did not converge within specified number of iterations
```

```
C            :4 - chosen pump is to small to match system,i.e. speed required to provide
C             flow is greater than maximum pump speed
C******************************************************************

    IMPLICIT NONE
    INTEGER INFO,COUNTER,MAXIT,LOOP,FAIL
    INTEGER NSTORE,IAV,ISTORE
    REAL TIME,XIN,OUT,T,DTDT,PAR,S
    REAL TL,N,Q,H,ETA,EPS,RHOG,PHYD,PMECH,QNOM,HOFF
    REAL NOFF,QOFF,ETAOFF,NREF,POFF,TOFF,TREQ
     REAL X(2),XNEW(2),J(2,2),JINV(2,2),F(2)
      REAL S1,S2,S3,S4,S5,S6,S7,S8
     REAL A1,B1,C1,E1,F1,G1,H1,PI,CONST
    REAL DET,DELTA1,DELTA2,NINIT,QINIT,NMAX
    DIMENSION XIN(1), OUT(8), PAR(6), INFO(10)

    COMMON /STORE/ NSTORE,IAV,S(5000)
     COMMON /COEF/ S1,S2,S3,S4,S5,S6,S7,S8,CONST
C-------------------------------------------------------------
C**** First call of component
    IF (INFO(7).EQ.-1) THEN
     INFO(6)=8
     INFO(9)=0

C****   storage allocation: speed and flowrate from previous
C****   iteration are stored in the S-array
C****   S(ISTORE) -- NOLD
C****   S(ISTORE+1) -- QOLD        INFO(10)=2
      CALL TYPECK(1,INFO,1,6,0)
      ISTORE=INFO(10)
      EPS=1.e-4
      MAXIT=100
C****   Set parameters
C****   System curve
      S3=PAR(1)
      S4=PAR(2)
     NREF=PAR(3)
     NMAX=PAR(4)
         POFF=PAR(5)
     QNOM=PAR(6)
C****   Coefficients for the equations
C****   Pump characteristics (obtained from regression)
         CALL HEADPOLYNOMIAL (A1,B1)
C****   Efficiency characteristics
         CALL EFFPOLYNOMIAL(E1,F1,G1,H1)

C****   Const=specific weight of water(rho*g) divided by 2*pi
      PI=ACOS(-1.)
         RHOG=9819.8
      CONST=RHOG/2./PI
```