Analysis of

Combined Power Systems

General description of software

Ingemar Mathiasson

November 2006

Department for Energy and Environment

Division of Electric Power Engineering

Chalmers University of Technology

Contents

1 INTRODUCTION

The following document deals with a method for analyzing the performances of large electric power systems with a base structure according to Figure 1. In this power system there are two types of power producers:

Type 1: The combination of Hydro-Wave-generators (HW)

Type 2: The combination of Wind-Sun-generators (WS)

The HW-generators are connected to the transmission grid (T) and to the international grid (I). The WS-generators are connected to the distribution grids (D) and to the consumer grids (C).

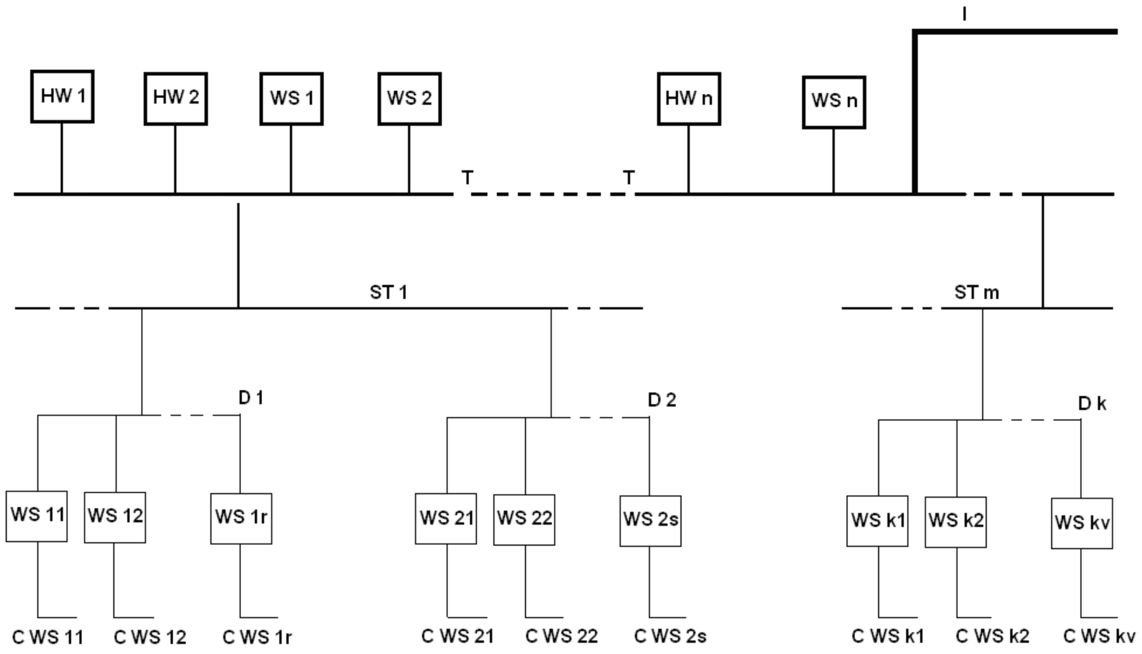

Figure 1 The base structure of the power system that could be analysed with the simulation system "*Combined_Power_system*"

<u>The following abbreviations have been used in Figure 1</u>:

HW a: Power production with the combination Hydro and Wave (unit number a)

WS b: Power production with the combination Wind and Sun (unit number b)

I: International grid

T. Transmission grid (voltage >130 kV)

ST c: Subtransmission grid (region grid) (voltage >10 kV to 130 kV) (unit number c)

D d: Distribution grid (voltage 10 kV) (unit number d)

C WS de: Consumer grid connected to distribution grid d (unit e)

<u>This paper only deals with the WS-system (Wind/Sun-system)</u>.

Figure 2 illustrates the structure of the power producers "*WS*" (Wind/Sun) and their connections.

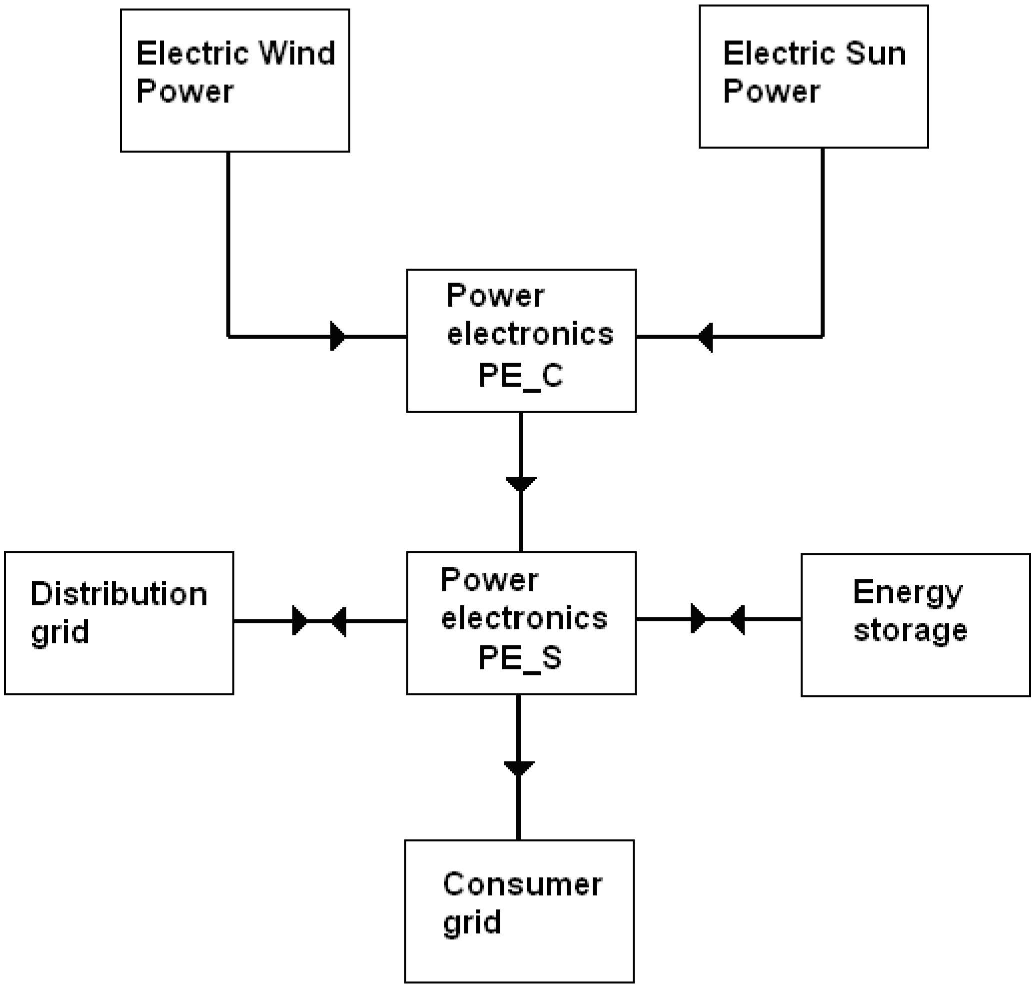

Figure 2 Basic structure of the power producers "*WS*" (Wind/Sun) and their connection. For further comments see Table 1

Block name in Figure 2	**Function**
Electric Wind Power	An optional number of wind turbines, including power electronics, that produce electric power
Electric Sun Power	An optional area of sun panels, including power electronics, that produce electric power
Power electronics PE_C	Power electronics for connection of *Wind power* and *Sun power* to a common power output
Power electronics PE_S	Power electronics for switching the power flow between the different system units
Distribution grid	Distribution grid for import/export of electric power
Energy storage	Energy storage for storing of surplus energy alternatively for delivering (to consumer grid) of deficit energy. When the storage is below a specified minimum level deficit energy is imported from the distribution grid. If the storage exceeds a specified maximum level surplus energy is exported to the distribution grid
Consumer grid	The local load of the power system

Table 1 Some comments regarding blocks according to Figure 2

2 PROGRAM STRUCTURE

The simulation programs are built up according to Table 2:

Program Block	**Function**
Main program:	
Combined_system	Simulate an electric power system according to Figure **2**. The main program organize the information flow between a number of subroutines. See below
Subroutines	
Wind_make	Generate a wind speed file
Wind_turbine	Generate electric power from one or a number of wind turbines as a result of the wind speed file
Extinction_make	Generate an extinction vector
Sun_intensity	Generate a radiation vector as a result of the extinction vector and the sun position relatively the solar cells panels
Sun_panel_generator	Generate electric power from a number of solar cells panels as a result of the radiation vector
Load_make	Generate a load vector (consumer grid)
Connect_Gen_load	Connection of a loading grid to the generators (wind and sun). Generate a gross power vector and a net power vector
Battery_Distribution	Connection of the combination *battery system - distribution grid* to the combination *consumerg grid - generators (wind and sun)*
Power_evaluate	Evaluate the result of a simulation process corresponding to *Mode 1* (energy balance)

Table 2 The structure of the simulation program for the power system analysis

3 MAIN PROGRAM

3.1 Principle

The simulation program simulates the process of a combined electric power system, where wind and sun are energy sources for electric power generation. The total system is built up by the following parts:

- Wind speed generation
- Extinction coefficient generation
- Sun irradiance generation
- Wind speed to electricity generation
- Sun irradiance to electricity generation
- A loading grid (consumer grid)
- Energy storage
- Distribution grid (for export/import)

The simulation could be processed for an optional number of simulation sequences. A number of 50 to 100 sequences is recommended.

3.2 Input parameters

The input parameters of the routine are specified in Table 3.

Parameter name	**Purpose**
N_sim_turns	Number of simulation sequences (turns) for a total simulation process
Sim_step_sec	Time interval in seconds per simulation step (60 is a standard value)
Sim_step_total	The total number of simulation steps per sequence (Sim_step_total = 43200 corresponds to a simulation sequence over a time of 30 days if Sim_step_sec = 60)
P_wind_file	Name of a file to store the electric wind power vector (string). The vector corresponding to the last simulation sequence is stored

P_sun_file	Name of a file to store the electric sun power vector (string). The vector corresponding to the last simulation sequence is stored
Fig_show	A choise whether plotting of diagrams corresponding to the first simulation sequence should be performed: Fig_show = 1 Plotting will be done Fig_show $\neq 1$ Plotting will not be done

Table 3 Input parameters for main program "Combined_Power_System"

3.3 Output parameters

The output parameters of the routine are specified in Table 4.

Parameter name	**Purpose**
Parameters with the following names are calculated: • time_vector • v_wind_vector • P_wind_el_vector • Ext_T_vector • Radiation_vector • P_sun_el_vector • P_load_vector • P_generation_vector • P_buffer_vector • battery_charge_vector • P_rel_batt_vector • P_rel_distr_vector • E_import_vector • E_export_vector	The following parameters are plotted vs time: • Wind rate • Electric wind power • Extinction coefficient • Sun irradiance at solar cells • Electric sun power • Loaded power (consumer grid) • Total generated power (wind + sun). I.e. "gross power" • Net power (generated power – consumer grid load) • Battery charge • The quotient between net power and maximum allowed charge power to battery • The quotient between net power and maximum allowed power to distribution grid • Imported power • Exported power
sim_turn	Current simulation sequence is displayed
P_wind_file	See Table 3
P_sun_file	See Table 3

Table 4 Output parameters for main program "Combined_Power_System"

4 SUBROUTINES

4.1 Wind_make

4.1.1 Principle

Wind_make is a program function (subroutine) with the purpose to generate a wind rate vector. A new vector is generated for each new simulation sequence. For further information see [1]. The wind rate values, in the following named v_wind, are generated as results of 1) weather variations and 2) turbulences. The total v_wind = Level_W + Level_T, where Lewel_W is a result of weather and Level_T is a result of turbulence. See Figure 3.

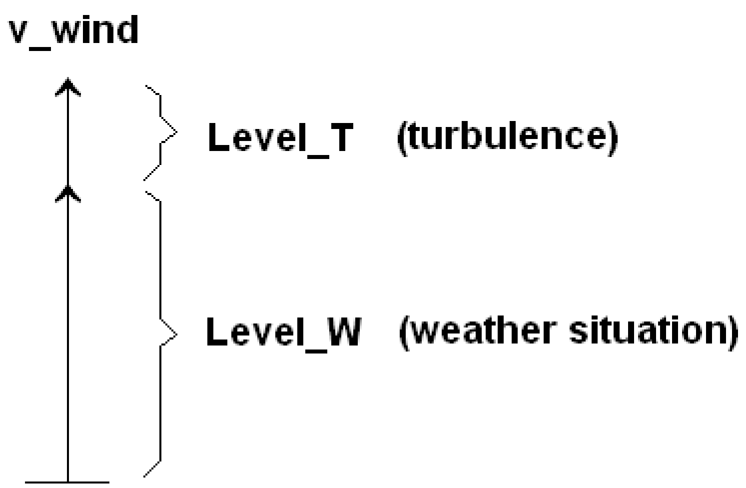

Figure 3 The wind rate is built up by two components, Level_W and Level_T

The simulation sequence consists of an optional number of simulation steps (Sim_step_total). The sequence is divided into a number of W-cycles, where each cycle is characterized of a “constant” weather situation. See Figure 4.

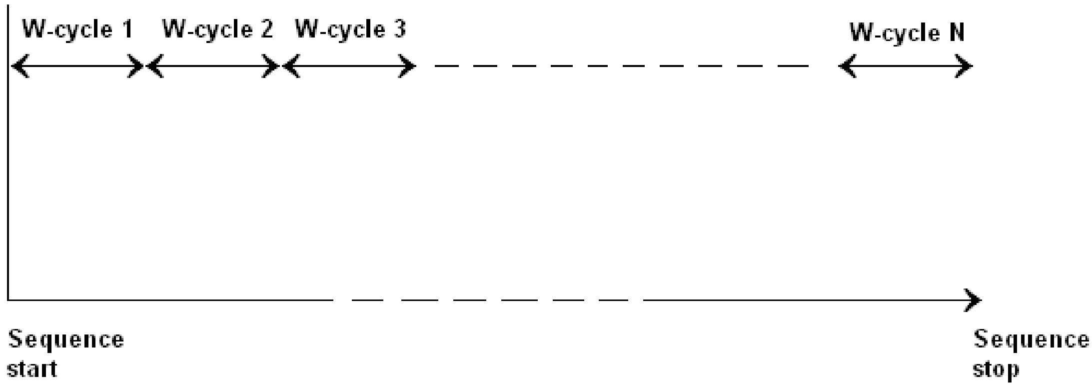

Figure 4 A total simulation sequence consists of a number (N) of W-cycles each of them representing a sertain weather situation

As mentioned above, the W-cycle contributes with a dominating “base” component to the wind speed. This base component is named Level_W. This is stochasticly generated by a “Weibull distribution” according to:

Equation 1:

Level_W = W(A,B)

Where W is a Weibull process and A respectively C are the “Weibull parameters”.
A new generation is performed for every W-cycle.

There is a “soft” transition from one W-cycle to another. That means that the final valuel of Level_W is on hand not until 50 % of the time for the W-cycle in question.

The number of simulation steps in a W-cycle, *Sim_step_W_total*, is stochasticly generated in two steps according to:

<u>Step 1</u>

Equation 2:

Sim_step_W_total_prel = N(Sim_step_W_My ,Sim_step_W_Sigma)

Where *Sim_step_W_total_prel* is a first preliminary number of simulation steps, *N* is a normal process and *Sim_step_W_My* and *Sim_step_W_Sigma* are input parameters corresponding to mean value respectively standard deviation of simulation steps per W-cycle.

Step 2

Sim_step_W_total_prel is then adapted to the statistic mean value of simulation steps per T-cycle, *Sim_step_T_My*, according to:

Equation 3:

T_cycles_total = ceil(Sim_step_W_total_prel / Sim_step_T_My),

Where ceil rounds the argument to the nearest integer upwards.

Equation 4:

*Sim_step_W_total = T_cycles_total * Sim_step_T_My*

Where *Sim_step_W_total* is the final value of simulation steps in the W-cycle in question.

Each W-cycle consists of a number of T-cycles. See Figure 5.

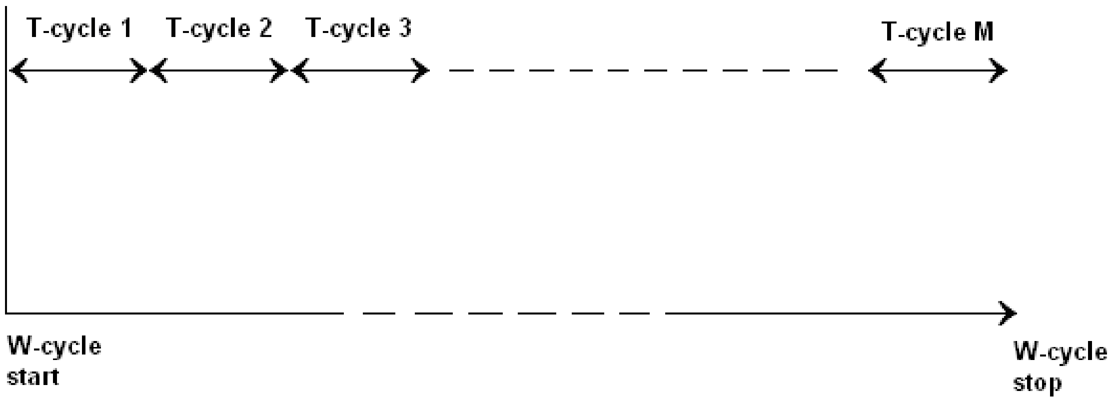

Figure 5 Every W-cycle is divided into a number (M) of T-cycles each of them representing a sertain turbulence situation

Each T-cycle depends on an individual turbulence situation, that is varied from T-cycle to T-cycle. The contribution, Level_T , to the total wind speed, v_wind (see above), is generated by a "Normal distribution" according to:

Equation 5:

Level_T = N (Level_T_My , Level_T_Sigma) where *Level_T_My* and *Level_T_Sigma* are input parameters corresponding to mean value respectively standard deviation of turbulence contribution. *Level_T_My* is normaly zero as the turbulence normaly is fluctuating around the zero level.

The generated value of Level_T, is linearly distributed during the first half of the T-cycle. During the second half of the T-cycle the level returns to zero. See Figure 6.

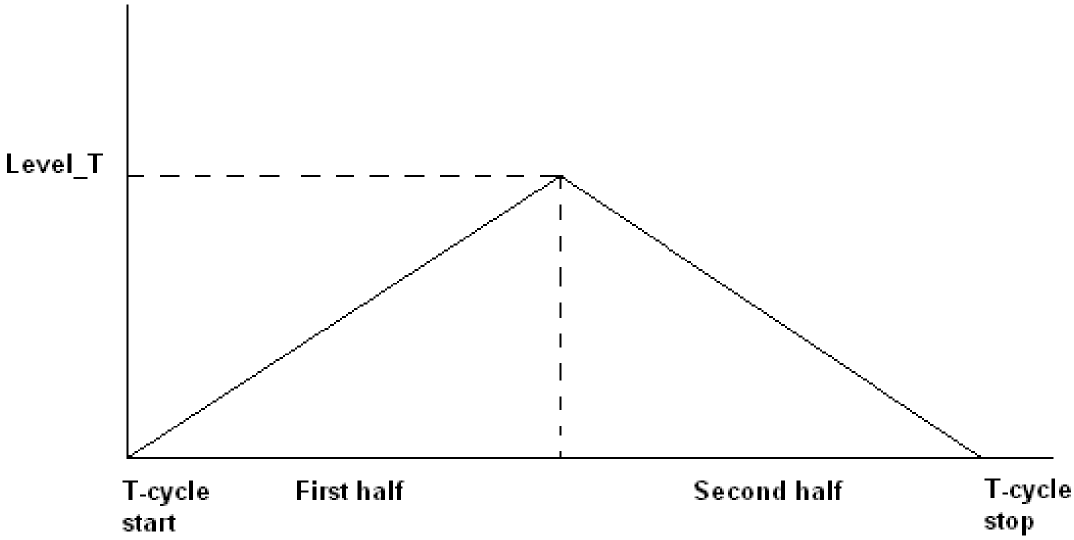

Figure 6 Distribution of Level_T during the T-cycle

The number of simulation steps for a single T-cycle, *Sim_step_T_total*, is stochasticly generated by a “Normal distribution” according to:

Equation 6:

Sim_step_T_total = N (Sim_step_T_My, Sim_step_T_Sigma)

Where *Sim_step_T_My* and *Sim_step_T_Sigma* are input parameters corresponding to mean value respectively standard deviation of number of simulation steps for the T-cycles.

4.1.2 Input parameters

The input parameters to the routine are specified in Table 5.

Parameter name	Purpose
Sim_step_sec Input via function argument from Main Program	Time interval in seconds per simulation step (60 is a standard value)
Sim_step_total Input via function argument from Main Program	The total number of simulation steps per sequence (Sim_step_total = 43200 corresponds to a simulation sequence over a time of 30 days if Sim_step_sec = 60)
Sim_step_W_My	Mean value of the number of simulation steps per W-cycle (Sim_step_W_My = 4320 corresponds to a mean value of 3 days (3 times 24 hours) if Sim_step_sec = 60)
Sim_step_W_Sigma	Standard deviation of the number of simulation steps per W-cycle
Sim_step_T_My	Mean value of the number of simulation steps per T-cycle (Sim_step_T_My = 10 corresponds to a mean value of 10 minutes if Sim_step_sec = 60)
Sim_step_T_Sigma	Standard deviation of the number of simulation steps per T-cycle
A	Weibull parameter (scale parameter)
C	Weibull parameter (shape parameter)
Level_T_Sigma_proc	Standard deviation of Level _T in percent of Level_W
v_wind_H	Upper limit of the wind speed
v_wind_L	Lower limit of the wind speed
Wind_speed_file	The name of a *Wind speed file* (string) to store the wind speed vector and the above parameters in this table

Table 5 Input parameters for routine “Wind_make”

4.1.3 Output parameters

The output parameters from the routine are specified in Table 6.

Parameter name	**Purpose**
time_vector	Vector with time in hour (per simulation step)
v_wind_vector	Vector with wind rates (per simulation step)
Wind_speed_file	See Table 5

Table 6 Output parameters for routine “Wind_make”

4.1.4 Examples

Table 7 gives an example of used parameters in a simulation.

Parameter name	**Used value in the example**
Sim_step_sec	60
Sim_step_total	43200
Sim_step_W_My	4320
Sim_step_W_Sigma	1500
Sim_step_T_My	10
Sim_step_T_Sigma	3
A	7.0
C	2.0
Level_T_Sigma_proc	30.0
v_wind_H	20.0
v_wind_L	0.0
Wind_speed_file	'Wind_1'

Table 7 An example of used input parameters for routine “Wind_make”

A statistically example when using input parameters according to Table 7 follows in Figure 7 (the total sequence of 720 hours), Figure 8 (the first 24 hours) and Figure 9 (the first 2.4 hours).

Have a look at Figure 8 and Figure 9 it can be observed that the turbulence contribution is oscillating around a base level, “above named “Level_W”, of about 4.7 metres per second. This base level is a result of stochastic generation by a Weibull distribution with the parameters A = 7.0 (scale parameter) and C = 2.0 (shape parameter).

For every new W-cycle a new generation of “Level_W” is performed. The resulting variations of base level during the simulation could be seen in Figure 7.

In Figure 10 different Weibull density functions are illustrated as functions of different Weibull parameters.

The turbulence contribution is stochastically generated by a Normal (Gauss) distribution. In the current example there has been used a standard deviation (Level_T_Sigma_proc) of 30 %. That means that the standard deviation in question is 30 % of the current base level (“Level_W”), in this case 30 % of about 4.7. As we normaly suppose that the turbulence is symmetric around the base level, the stochastically mean level is zero. Figure 8 and Figure 9 illustrate the turbulence effect. The top-/bottom level of the triangles (that corresponds to the turbulence contribution), illustrated in Figure 9, is the level (“Level_T”) that is generated as a result of “Level_T_Sigma_proc”. Figure 11 gives some examples of Normal density functions with varying standard deviations; 20 %, 30 % and 50 % of “Level_W”. These examples assume a base level of 4.7 m/s (corresponding to the situation in Figure 9).

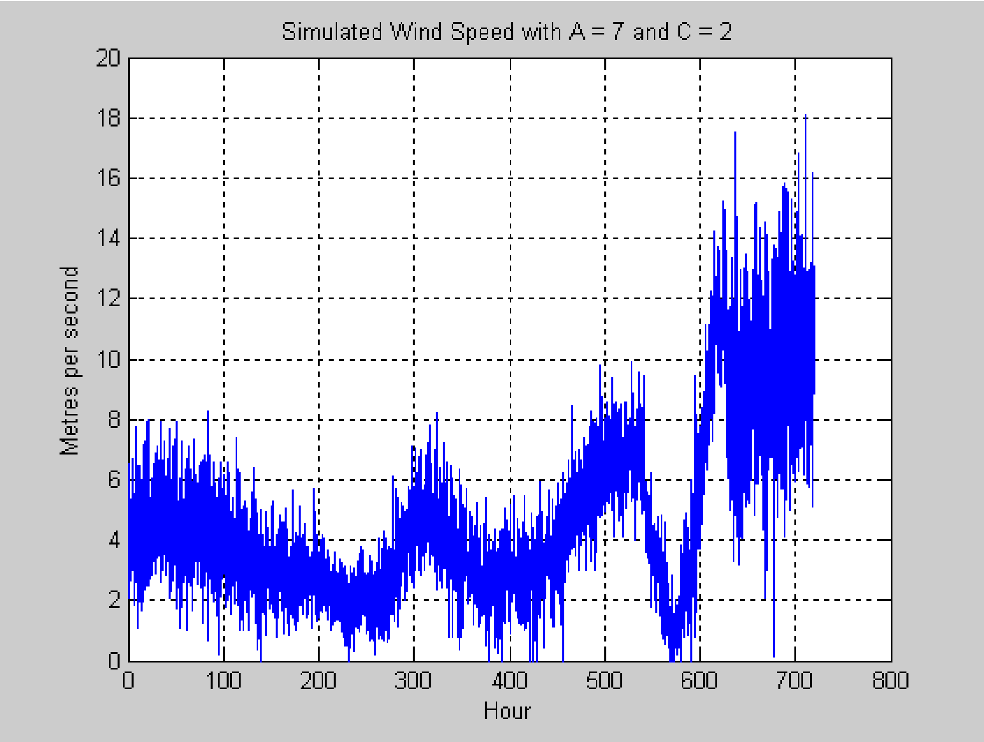

Figure 7 Simulated Wind speed for A = 7 and C = 2.
The total sequence is 720 hours

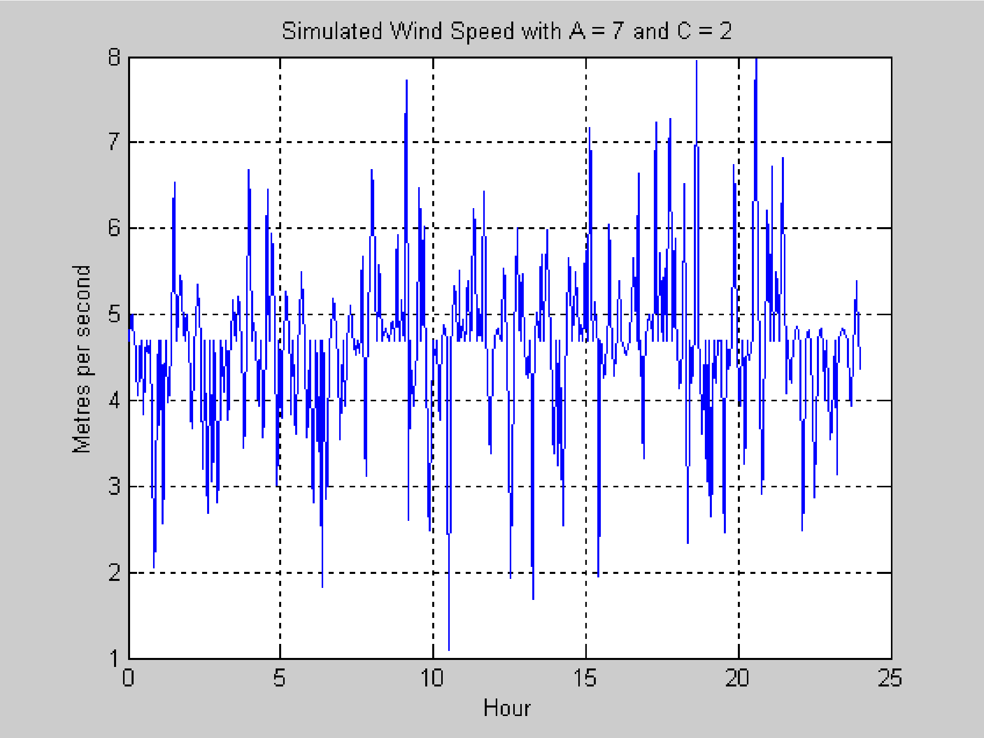

Figure 8 Simulated Wind speed for A = 7 and C = 2. The first 24 hours

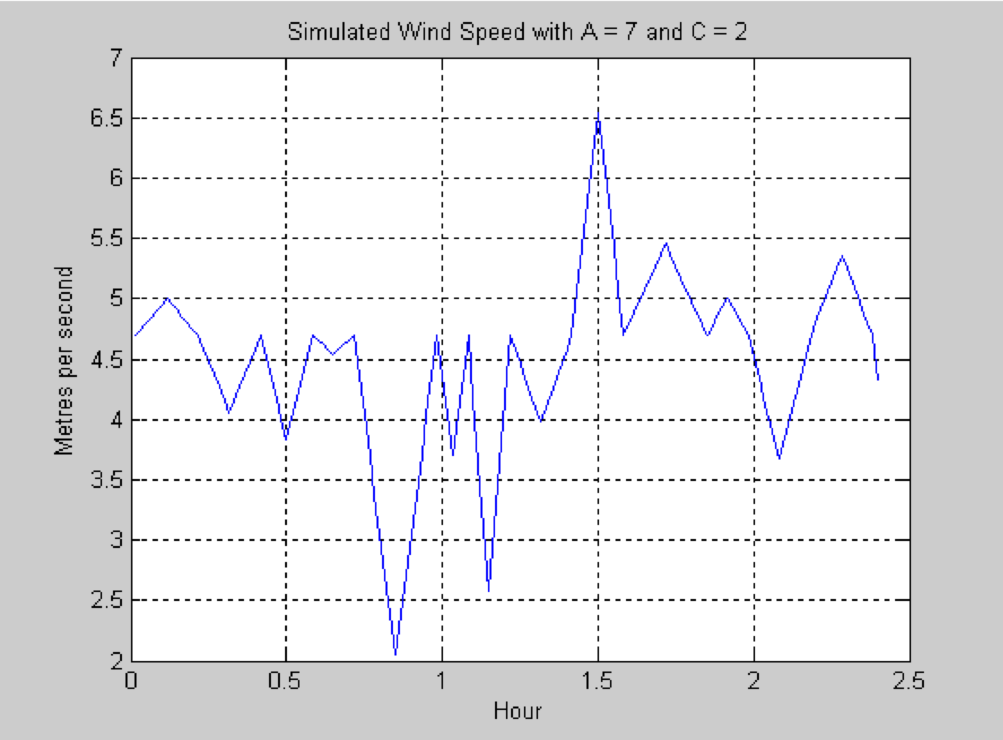

Figure 9 Simulated Wind speed for A = 7 and C = 2. The first 2.4 hours

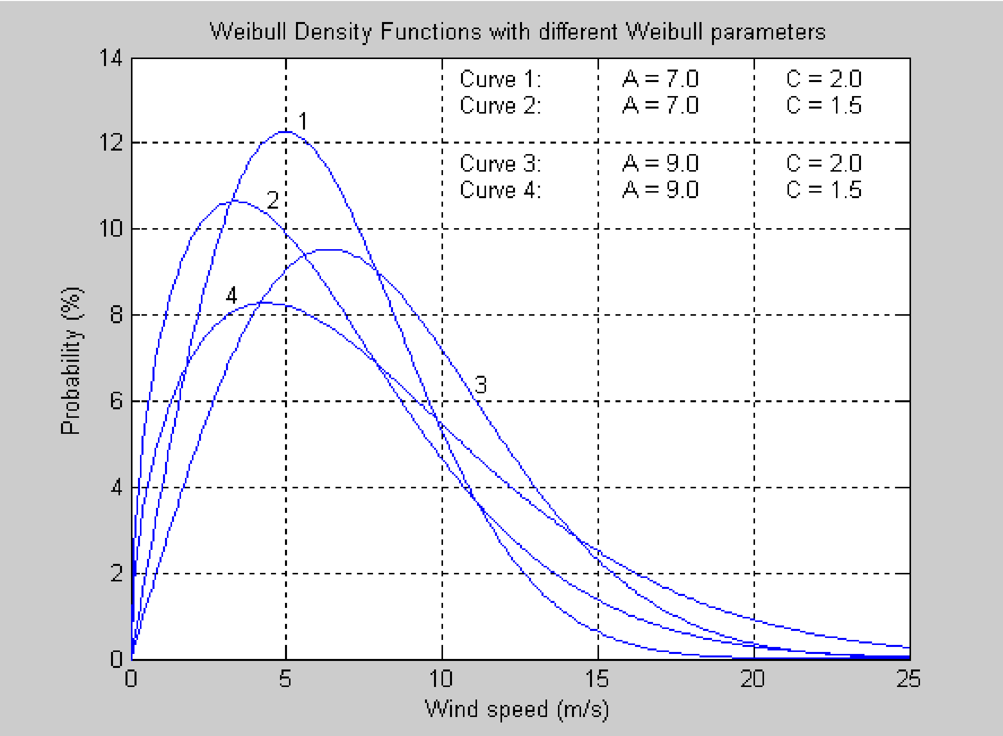

Figure 10 Examples of different Weibull Density Functions

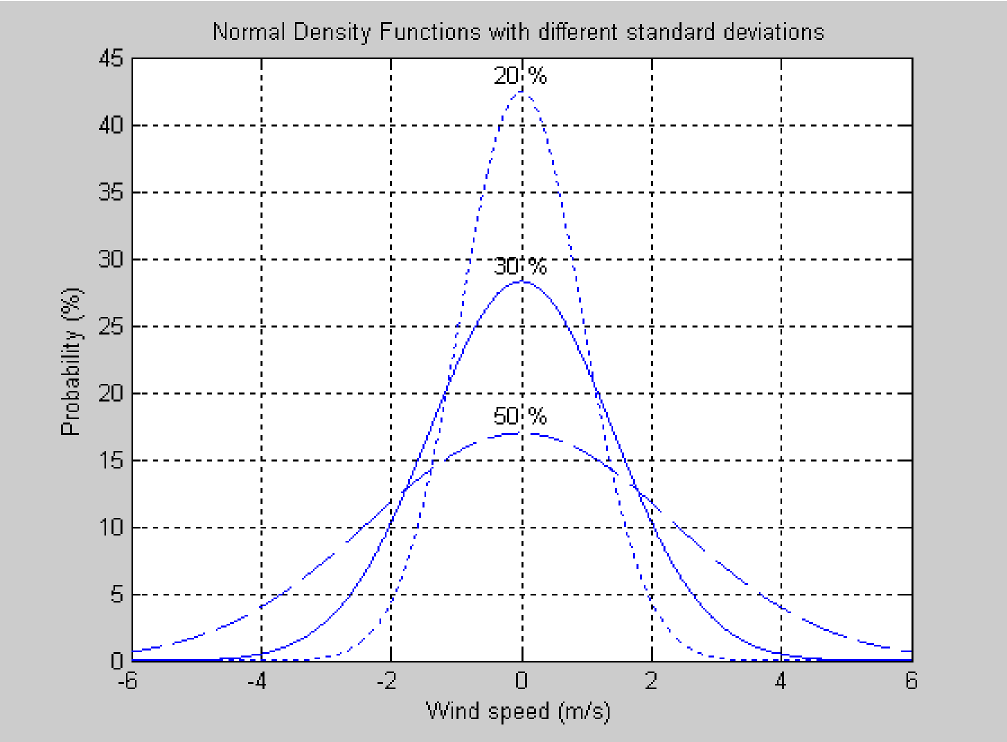

Figure 11 Normal density functions with varying standard deviations, relating to the parameter “Level_T_Sigma_proc” and the value of “Level_W” = 4.7 m/s. The 30 % curve corresponds to the chosed “Level_T_Sigma_proc” in the simulation according to the example

As mentioned above the length of the time intervals corresponding to each W-cycle and T-cycle are continuously and stochastically generated during the simulation. Figure 12 and Figure 13 illustrate the Normal density functions that are in questions for these processes.

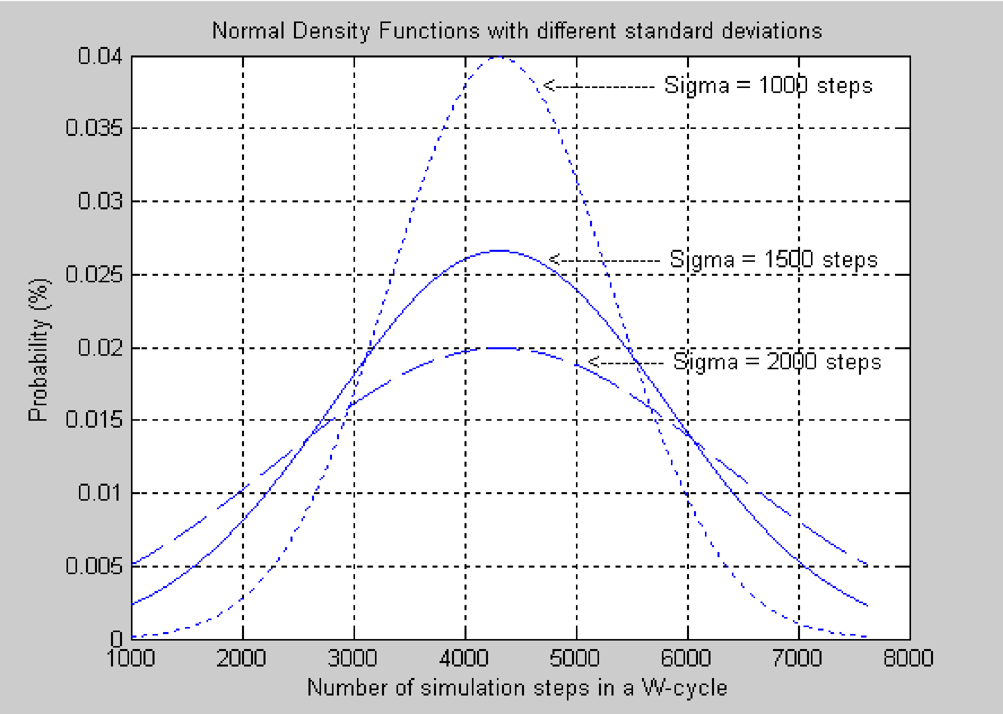

Figure 12 Normal density functions with varying standard deviations, relating to the parameter "Sim_step_W_Sigma". The "Sigma = 1500 steps" curve corresponds to the current the simulation in the example

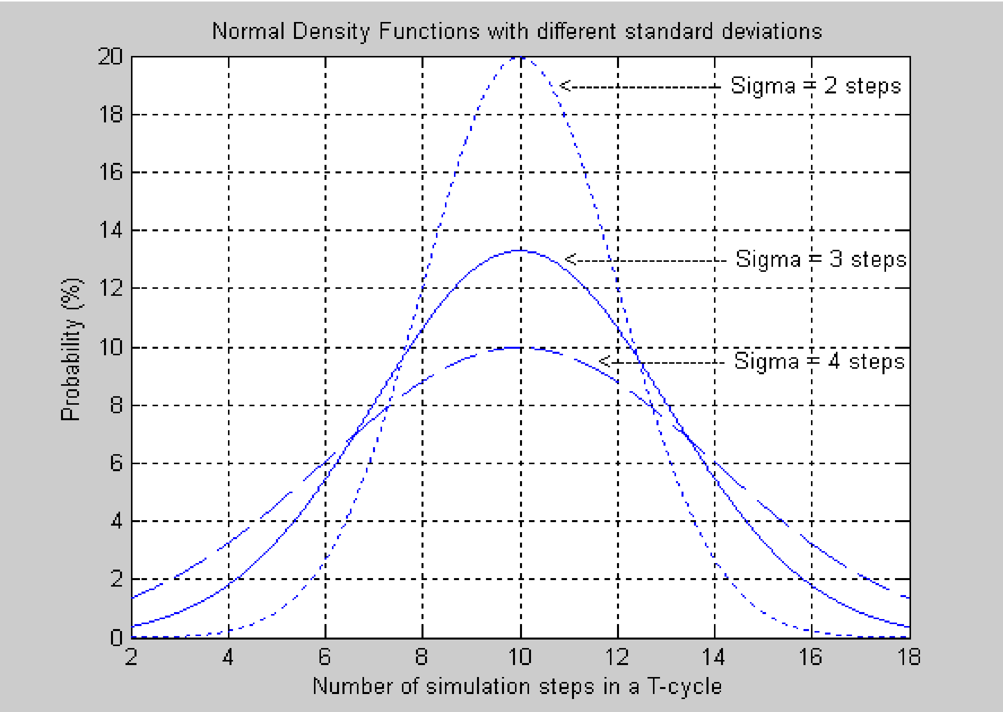

Figure 13 Normal density functions with varying standard deviations, relating to the parameter "Sim_step_T_Sigma". The "Sigma = 3 steps" curve corresponds to the current the simulation in the example

4.2 Wind_turbine

4.2.1 Principle

Wind_turbine is a program function (subroutine) with the purpose to simulate the electric power generation by **one** or **more** wind power turbine(s). The wind rate vector that is generated by the function Wind_make is used as one of the input parameters. To calculate the current wind power the following equation is used (see e.g. [4]):

Equation 7: $$Pw = Cp \cdot \frac{1}{2} \cdot \rho \cdot A \cdot V^3$$

Where

Pw: wind power (W)

Cp: power koefficient

ρ: the air density ($\frac{kg}{m^3}$)

A: rootor sweeping area (m^2) as a result of rootor diameter according to:

Equation 8: $$A = D_R{}^2 \cdot \frac{\pi}{4}$$

D_R: rootor diameter (m)

V: wind speed ($\frac{m}{s}$)

ρ is calculated according to:

Equation 9: $$\rho = \frac{1.293}{1 + 0.00367 \cdot Tair} \cdot \frac{Pair}{1013}$$

Where

$Tair$ is the air temperature in $°C$

And

$Pair$ is the air pressure in $mbar$

The power coefficient Cp is a function of a parameter λ, the so called “tip speed ratio” (%) according to:

Equation 10: $$\lambda = \frac{Vt}{V}$$

Where

Vt is the wing tip speed ($\frac{m}{s}$)

The relation between Cp and λ is in the routine calculated by a polynom of grade 5 according to:

Equation 11:

$$Cp = 1.142515 - 1.253909\lambda + 0.478158 \cdot \lambda^2 - 0.07554 \cdot \lambda^3 + 0.005426 \cdot \lambda^4 - 1.4623 \cdot 10^{-4} \cdot \lambda^5$$

This is a result of a polynom adaptation of the relation between Cp and λ that is valid at Chalmers Test Wind Turbine at Hönö. For measured values see [5]. Figure 14 shows the measured and adapted curves.

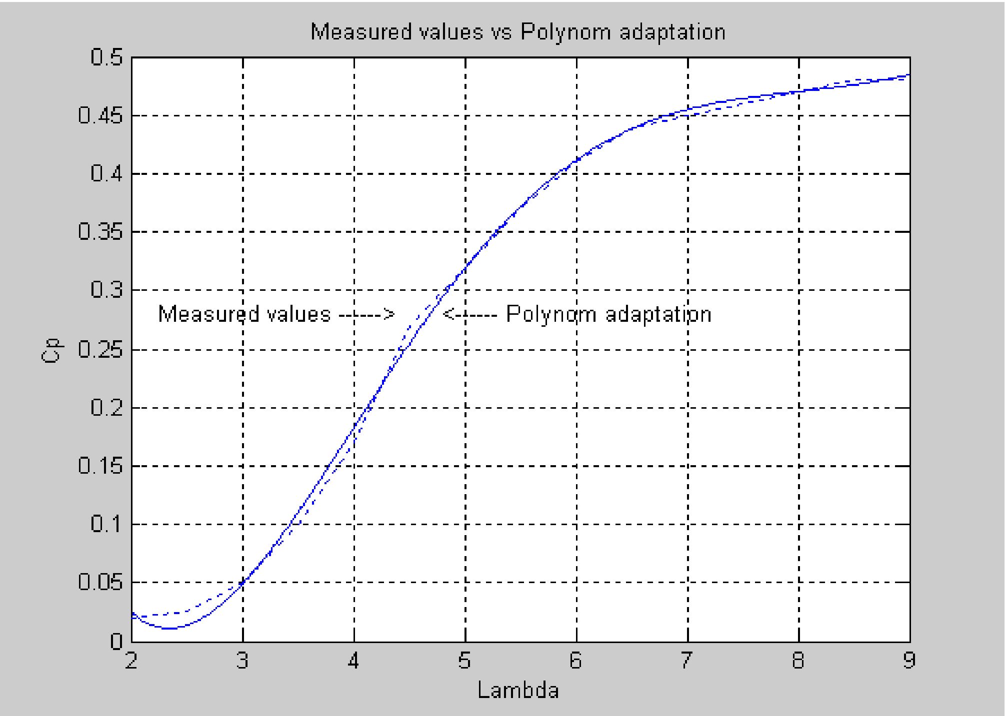

Figure 14 The relation between Cp and λ at Chalmers Test Wind Turbine at Hönö. The figure shows the measured respectively the polynom adapted relations

The routine presumes that the turbine, for a given wind speed, will regulate the rotation speed to get that λ that results in maximum Cp - value.

The rotation speed is limited by an optional input parameter. That means that the turbine regulation regarding λ only will adjust the rotation speed to give Cp - max up to a certain wind speed. If this wind speed is exceeded the λ - value will be less than the value that corresponds to Cp - max.

Figure 15 illustrates an example of the relation between wind speed and rotor speed. The rotor speed is limited at 85 rpm. Below this limit the control system adjust the rotor speed to get a λ -value that results in Cp - max. When the rotor speed limit is exceeded the λ -value decreases continuously according to Figure 16. This will on the other hand decrease the Cp - value (following the relation according to Figure 14) as is shown in Figure 17.

Figure 18 presents the electrical output power as a function of current wind speed. The electrical output power could be limited by an input parameter. The figure gives two examples. The solid curve represents the case with a limit of 35 kW. The dashed part gives the power without any limit.

Other input parameters that have been used in the current examples is (see also paragraph 4.2.2.):

Number of wind turbines: 1

λ_{REF} : 9.0

Where λ_{REF} is the λ -value that gives Cp - max

Mechanical and electrical efficiency η : 0.85

Where η is defined according to the following:

Equation 12: $$P_E = P_W \cdot \eta$$

P_E : Electrical power

P_W : Wind power according to Equation 7

$Tair$: 15 $°C$ (see Equation 9)

$Pair$: 1013 $mbar$ (see Equation 9)

D_R : 13.5 m (see Equation 8)

<u>Maximum and minimum windrates</u>

- If the wind rate exceeds a certain defined maximum level (*v_wind_max*) the turbine(s) is (are) stopped, resulting in no output power.

- If the wind rate is lower than a certain defined minimum level (*v_wind_min*) the turbine(s) does (do) not produce any output power.

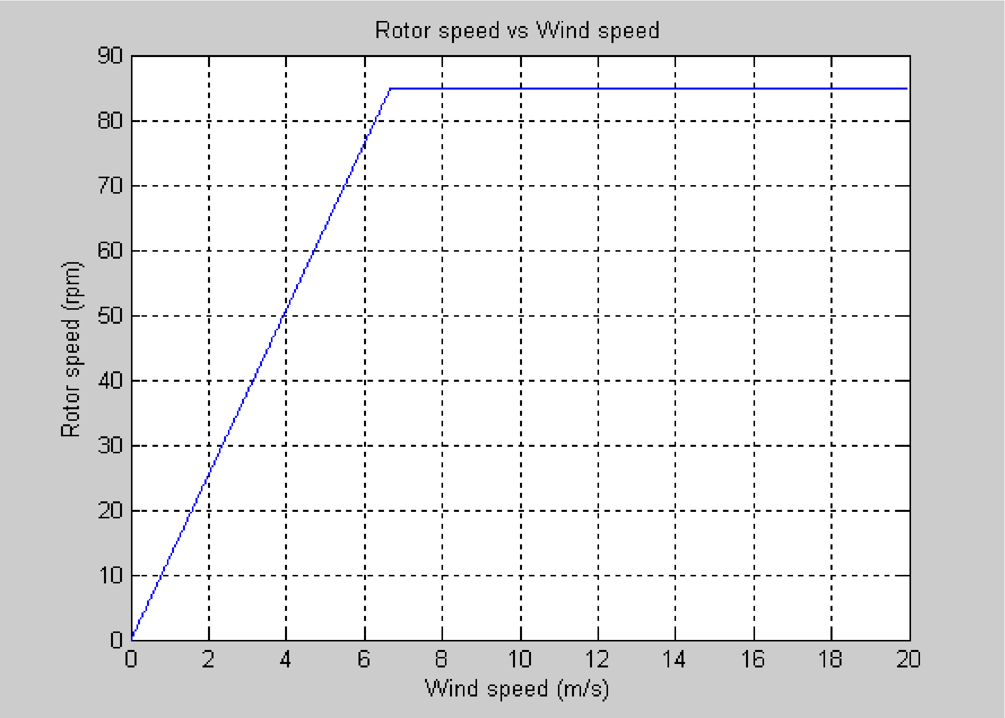

Figure 15	The rotor speed is limited. In this example at 85 rpm

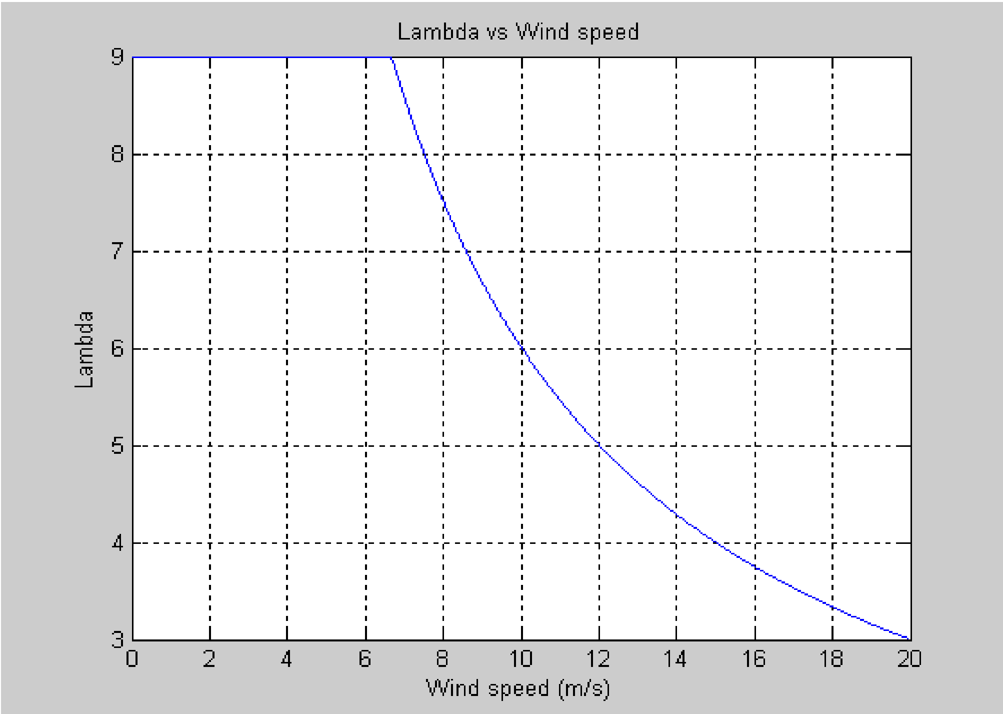

Figure 16 As the rotor speed is limited (85 rpm) the λ - value will decrease for rotor speeds that exceeds the limit

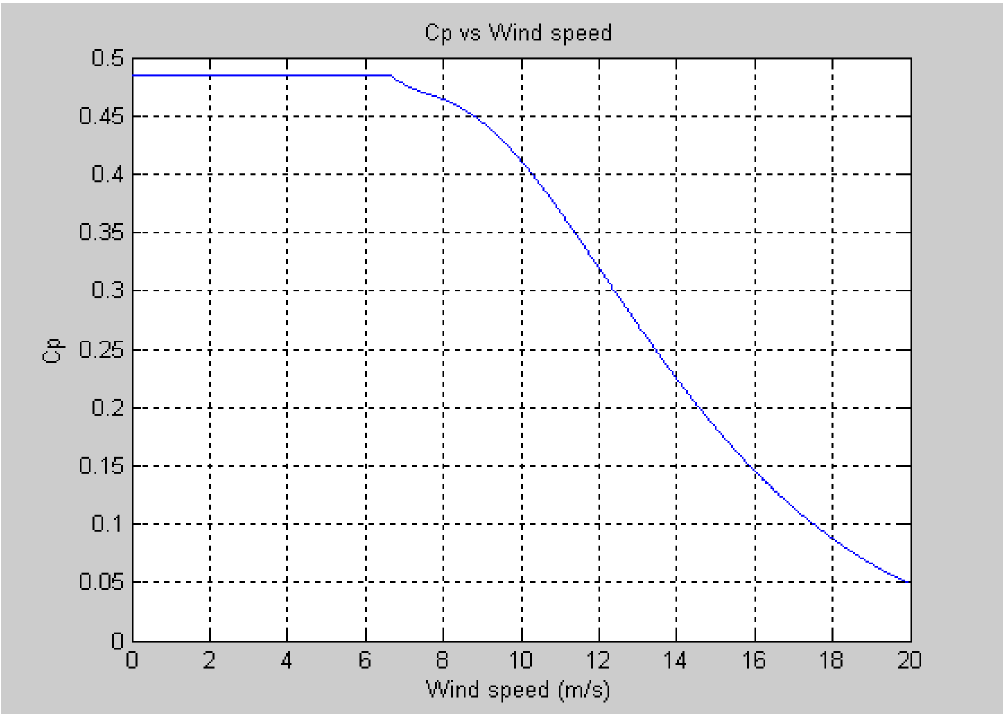

Figure 17 As λ is decreased for rotor speeds exceeding the rotor speed limit the consequence will be a decreasing Cp - value

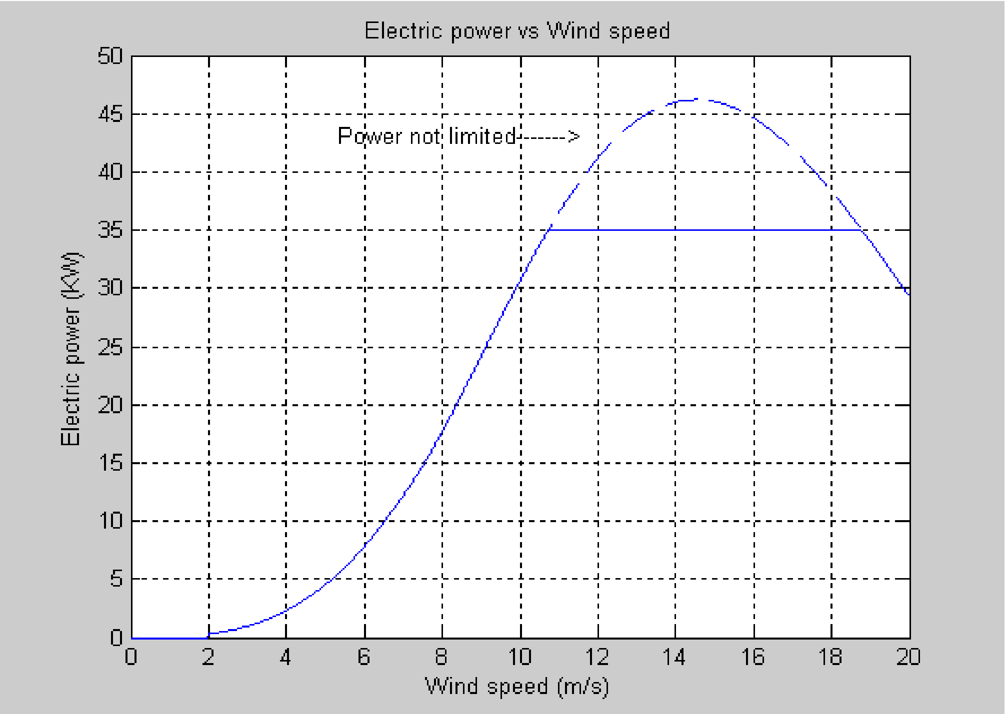

Figure 18 Resulted electric power as a function of wind speed. The dashed part of the curve corresponds to the circumstances when the electric power is not limited

4.2.2 Input parameters

The input parameters of the routine are specified in Table 8.

Parameter name	**Purpose**
v_wind_vector Input via function argument from Main Program	Vector with wind rates (per simulation step)
Turb_numbers	Number of wind turbines
lambda_ref	The λ -value that gives Cp - max
max_power	Limit for electrical output power from one turbine (W)
rotor_speed_max	Rotation speed limit for the turbine (rpm)
aeta	Mechanical and electrical efficiency. See Equation 12
t_air	The air temperature ($°C$). See Equation 9
p_air	The air pressure ($mbar$). See Equation 9
d_rootor	Rootor diameter (m)

v_wind_max	Maximum allowed vind rate (m/s) for function. If this wind rate is exceeded, the wind turbine(s) is (are) stopped
v_wind_max	Minimum wind rate (m/s) to result in power

Table 8 Input parameters for routine “Wind_turbine”

4.2.3 Output parameters

The output parameters of the routine are specified in Table 9.

Parameter name	**Purpose**
P_wind_el_vector Output via function argument to Main Program	Vector with electric wind power (per simulation step)

Table 9 Output parameters for routine “Wind_turbine”

4.2.4 Examples

Table 10 gives an example of used parameters in a simulation.

Parameter name	**Used value in the example**
Turb_numbers	2
lambda_ref	9.0
max_power	50 000
rotor_speed_max	85
aeta	0.85
t_air	15.0
p_air	1013.0
d_rootor	13.5
v_wind_max	25
v_wind_min	2

Table 10 An example of used input parameters for routine “Wind_turbine”

A statistically example when using input parameters according to Table 10 follows in Figure 19 (wind speed, a sequence of 24 hours) and Figure 20 (electrical output power, a sequence of 24 hours) respectively Figure 21 (wind speed, the first 2.4 hours of the sequence) and Figure 22 (electrical output power, the first 2.4 hours of the sequence).

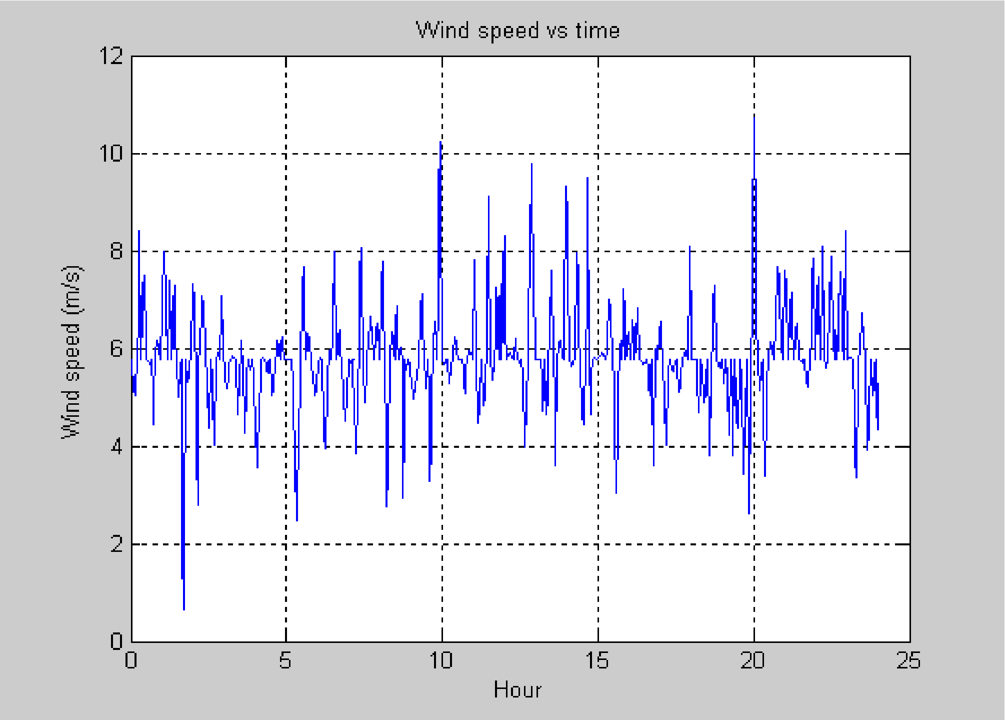

Figure 19 Wind speed vs time in the current example

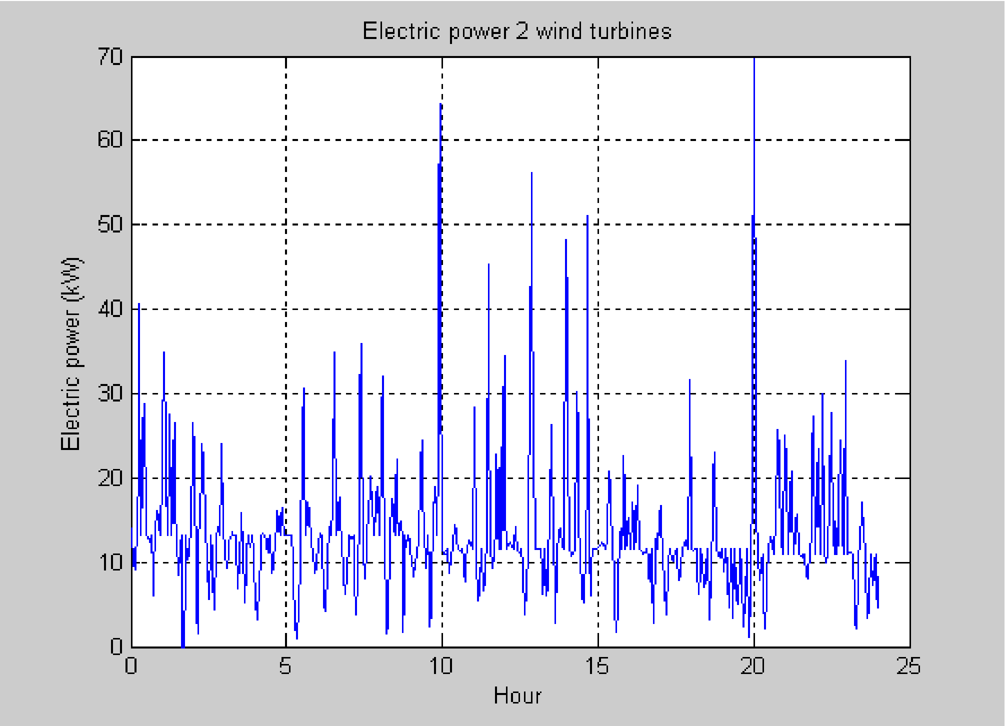

Figure 20 Electric power with wind speeds according to Figure 19 and Table 10

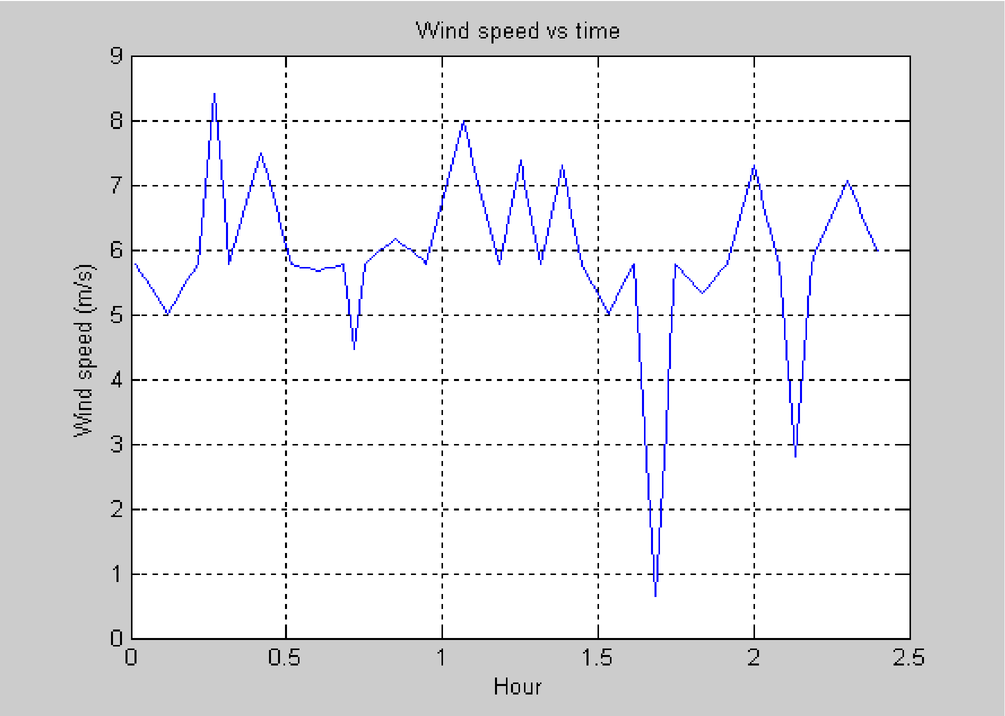

Figure 21 Wind speed vs time. The first 2.4 hours in the current example

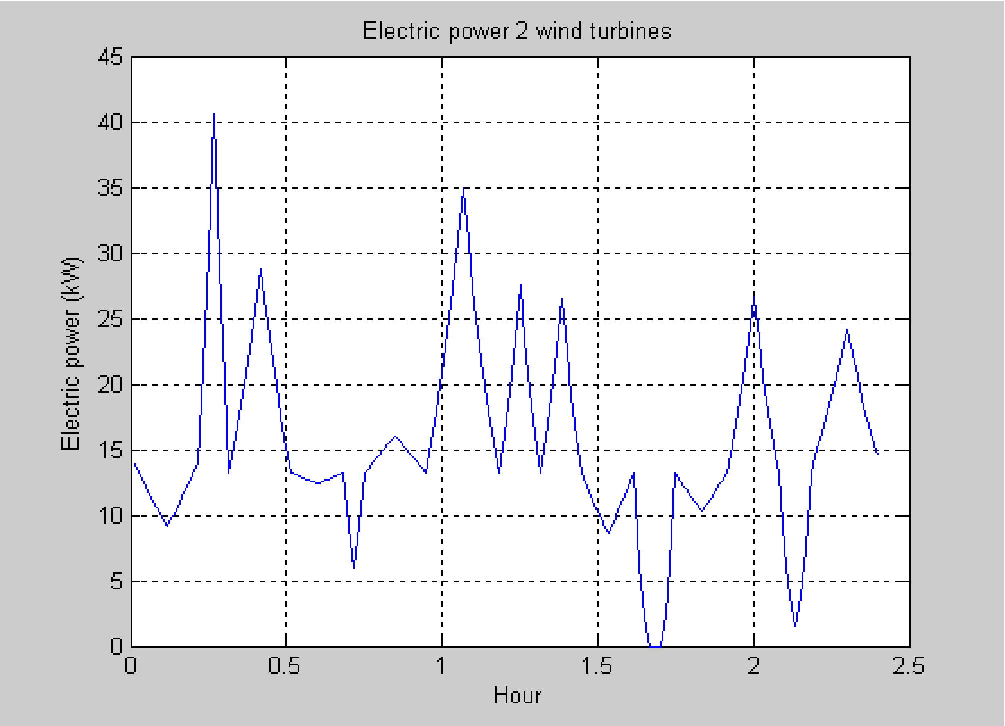

Figure 22 Electric power with wind speeds according to Figure 21 and Table 10

4.3 Extinction_make

4.3.1 Principle

Extinction_make is a program function (subroutine) with the purpose to generate an extinction vector. For definition of the parameter “*Extinction*” (or “*Extinction coefficient*”) see 4.4.1, Equation 37 and Equation 38. A new vector is generated for each new simulation sequence.

The extinction values, in the following named Ext_T, are generated as results of 1) low frequency variations and 2) high frequency variations. The total Ext_T = Ext_L + Ext_H, where Ext_L is the low frequency contribution and Ext_H is the high frequency contribution. See Figure 23.

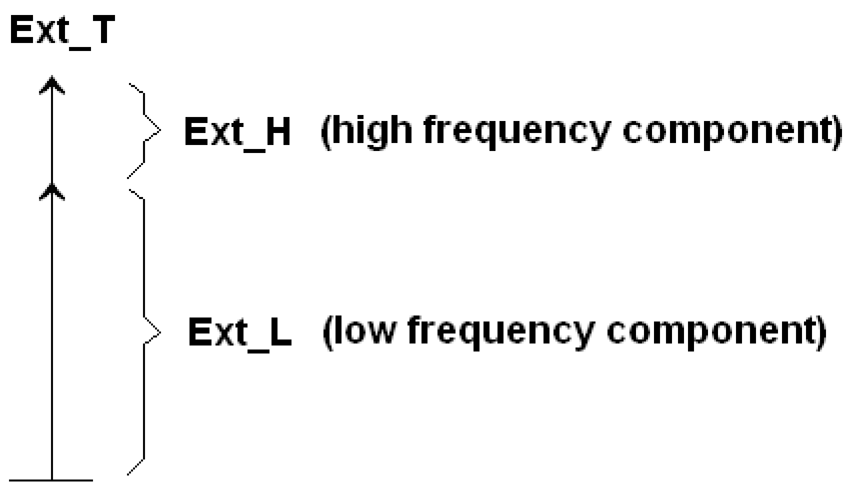

Figure 23 The extinction is built up by two components, Ext_L and Ext_H

The simulation sequence consists of an optional number of simulation steps (Sim_step_total). The sequence is divided into a number of L-cycles, where each cycle is characterized of a "constant" weather situation. See Figure 24.

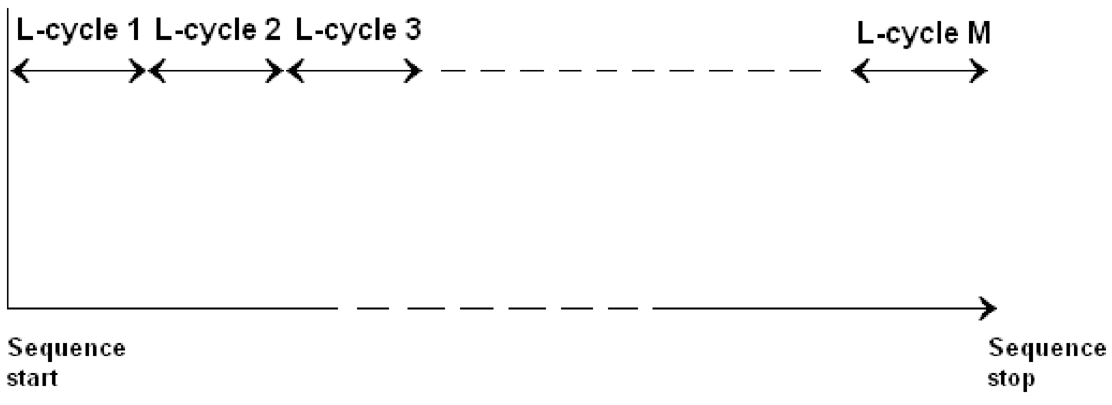

Figure 24 A total simulation sequence consists of a number (M) of L-cycles each of them representing a certain weather situation

Ext_L is stochasticly generated by a "Normal distribution" according to:

Ext_L = N(Ext_L_My,Ext_L_Sigma)

Where *Ext_L_My* and *Ext_L_Sigma* are input parameters corresponding to mean value respectively standard deviation of the low frequency component.
A new generation is performed for every new L-cycle.

There is a “soft” transition from one L-cycle to another. That means that the final value of Ext_L is on hand not until 50 % of the time for the L-cycle in question.

The number of simulation steps in an L-cycle, *Sim_step_L_total*, is stochasticly generated in two steps according to:

Step 1

Sim_step_L_total_prel = N(Sim_step_L_My ,Sim_step_L_Sigma)

Where *Sim_step_L_total_prel* is a first preliminary number of simulation steps, *N* is a normal process and *Sim_step_L_My* and *Sim_step_L_Sigma* are input parameters corresponding to mean value respectively standard deviation of simulation steps per L-cycle.

Step 2

Sim_step_L_total_prel is then adapted to the statistic mean value of simulation steps per T-cycle, *Sim_step_T_My*, according to:

H_cycles_total = ceil(Sim_step_L_total_prel / Sim_step_H_My),

Where ceil rounds the argument to the nearest integer upwards.

Finally:

*Sim_step_L_total = H_cycles_total * Sim_step_H_My*

Where *Sim_step_L_total* is the final value of simulation steps in the L-cycle in question.

Each L-cycle consists of a number of H-cycles. See Figure 25.

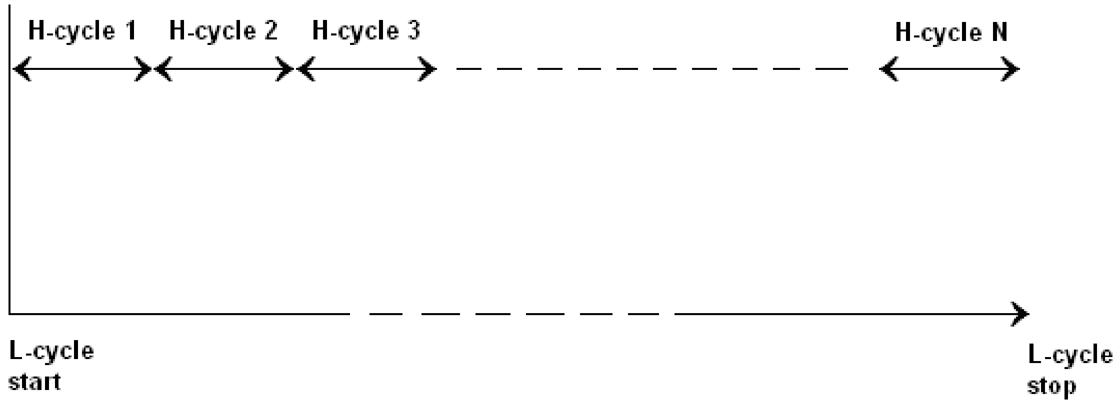

Figure 25 Every L-cycle is divided into a number (N) of H-cycles each of them representing a certain extinction variation

Each H-cycle depends on an individual extinction situation, that is varied from H-cycle to H-cycle. The contribution, Ext_H, to the total extinction, Ext_T (see above), is generated by a “Normal distribution” according to:

Ext_H = N (Ext_H_My , Ext_H_Sigma)

where *Ext_H_My* and *Ext_H_Sigma* are input parameters corresponding to mean value respectively standard deviation of high frequency contribution.

The generated value of *Ext_H* could be distributed according to two alternative methods:

<u>Alternative 1</u> (step distribution)

The total value of Ext_H is distributed direct at start of the H-cycle and change to a new value direct at start of the next H-cycle. See Figure 26.

<u>Alternative 2</u> (triangular distribution)

The value is linearly distributed during the first half of the H-cycle. During the second half of the H-cycle the level returns to zero. See Figure 27.

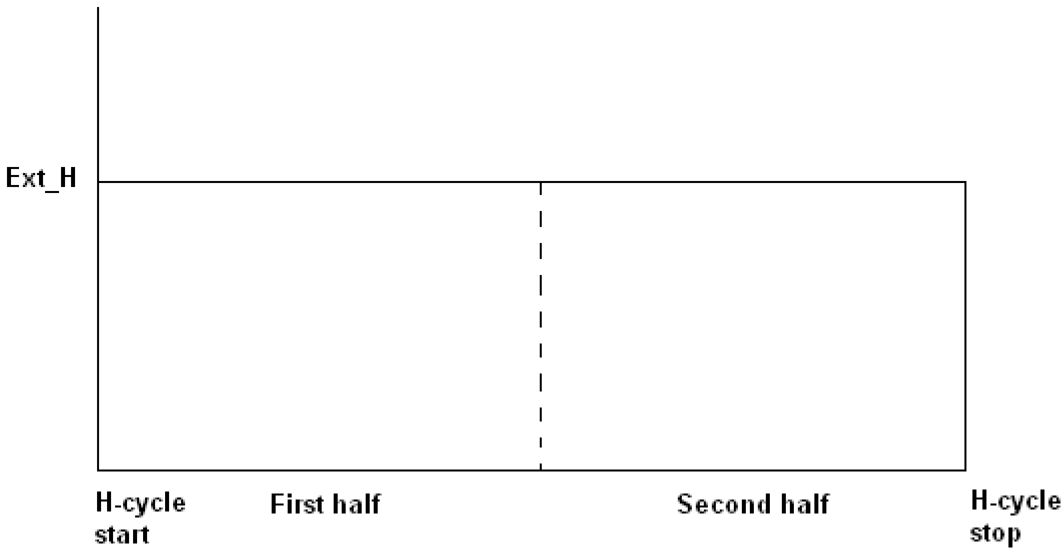

Figure 26 Distribution of Ext_H during the H-cycle at “Step distribution”

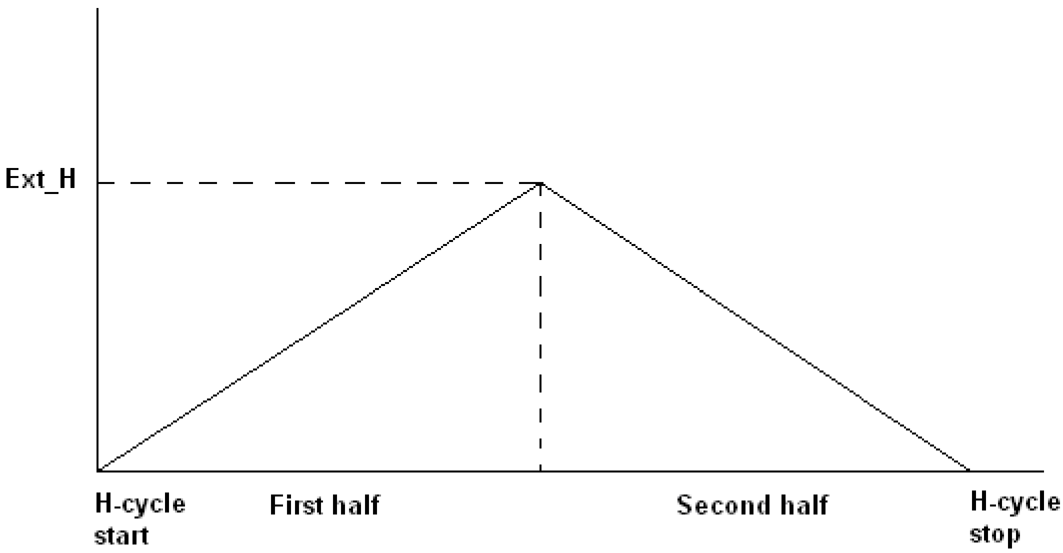

Figure 27 Distribution of Ext_H during the H-cycle at “Triangular distribution”

The number of simulation steps for a single H-cycle, *Sim_step_H_total*, is stochasticly generated by a “Normal distribution” according to:

Sim_step_H_total = N (Sim_step_H_My, Sim_step_H_Sigma)

Where *Sim_step_H_My* and *Sim_step_H_Sigma* are input parameters corresponding to mean value respectively standard deviation of number of simulation steps for the H-cycles.

As the H-cycles very often will be used to simulate temporary cloud variations, the H-cycles are to be repeated according to defined stochastic processes. These processes are controlled by a parameter, in the routine named *"H_limit"*. This parameter will, based on a "Normal distribution", result in defined possibilities of starting a H-cycle at a given time point. The principle to set the possibility of starting up a H-cycle (at a given time point) is explained by help of the example in Figure 28. The figure illustrates the so called "Standardized Normal Distribution". In the figure *"H_limit"* has been assigned the value 1.5. The possibility to start up a H-cycle at a given time point, in the following named P_H, corresponds to the blacked areas in the figure. In other words:

$$P_H(\text{H\_limit}) = 2 \cdot (1 - F(\text{H\_limit}))$$

Where

$F(x)$ is the integrated distributed function of a standardized Normal random variable.

H_limit = 1.5 results in P_H = 13.36 %.

Suppose that there (as an example) is a cloudiness of $\frac{2}{8}$. A good modeling of this regarding parameter *"H_limit"* is the following:

Cloudiness: $\frac{2}{8} = 0.25$

This corresponds to a possibility for cloud of about 25 %.

The inverse of $P_H(\text{H\_limit})$ = 25 % results in *H_limit* = 1.15

Following this principle for other values of the cloudiness results in the relation between *H_limit* and cloudiness according to Table 11.

Figure 29 shows P_H as a function of *H_limit.*

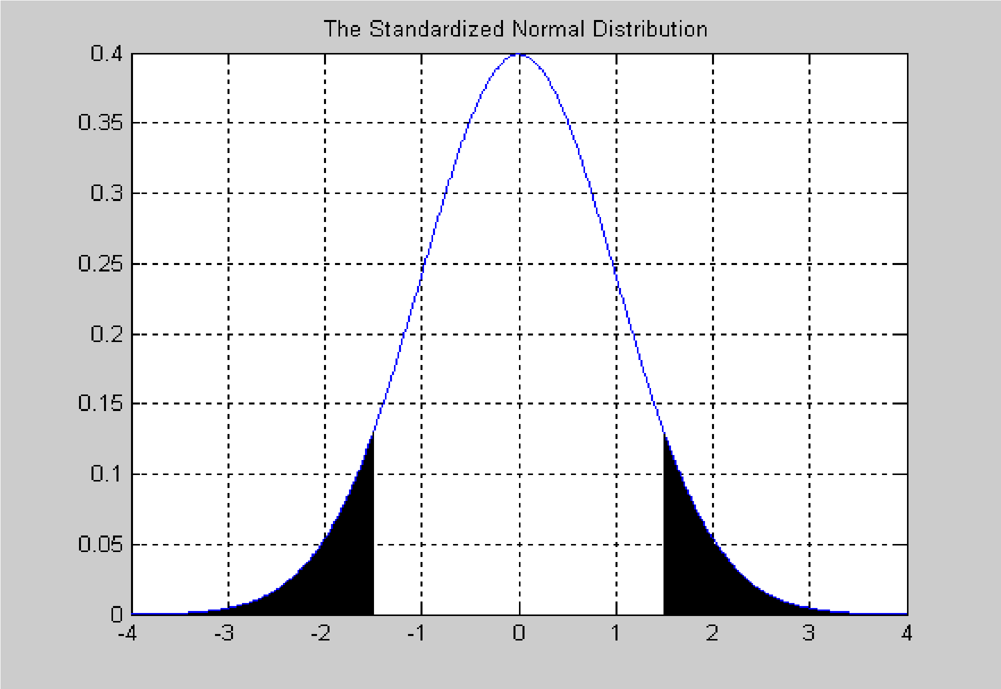

Figure 28 The parameter H-limit is in this example 1.5. P_H corresponds to the blacked areas, that is the same as 13.36 % of the total area

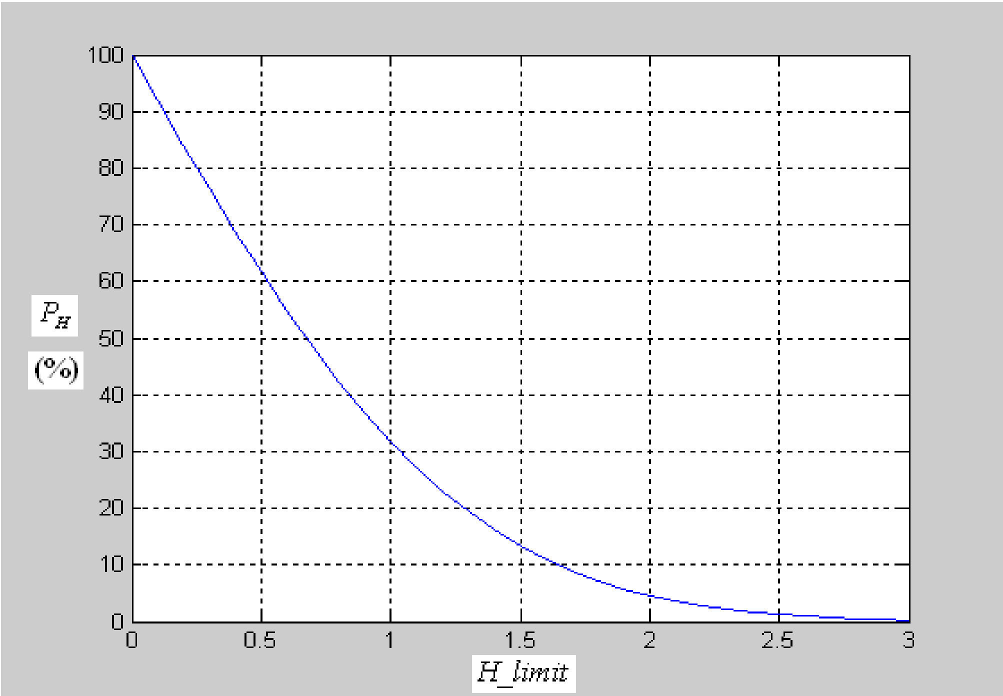

Figure 29 P_H as a function of *H_limit*

Cloudiness	P(cloud) (%)	*H_limit*
1/8	12.5	1.534
2/8	25.0	1.150
3/8	37.5	0.887
4/8	50.0	0.674
5/8	62.5	0.489
6/8	75.0	0.319
7/8	87.5	0.157
8/8	100	0

Table 11 *H_limit* as a function of cloudiness

For further information about the extinction model see [2].

4.3.2 Input parameters

The input parameters of the routine are specified in Table 12.

Parameter name	**Purpose**
Sim_step_sec Input via function argument from Main Program	Time interval in seconds per simulation step (60 is a standard value)
Sim_step_total Input via function argument from Main Program	The total number of simulation steps per sequence (Sim_step_total = 43200 corresponds to a simulation sequence over a time of 30 days if Sim_step_sec = 60)
Sim_step_L_My	Mean value of the number of simulation steps per L-cycle (Sim_step_L_My = 4320 corresponds to a mean value of 3 days (3 times 24 hours) if Sim_step_sec = 60)

Sim_step_L_Sigma	Standard deviation of the number of simulation steps per L-cycle
Sim_step_H_My	Mean value of the number of simulation steps per H-cycle (Sim_step_H_My = 10 corresponds to a mean value of 10 minutes if Sim_step_sec = 60)
Sim_step_H_Sigma	Standard deviation of the number of simulation steps per H-cycle
Ext_L_My	Mean value of the low frequency contribution of the extinction coefficient
Ext_L_Sigma	Standard deviation of the low frequency contribution of the extinction coefficient
Ext_H_My	Mean value of the high frequency contribution of the extinction coefficient
Ext_H_Sigma	Standard deviation of the high frequency contribution of the extinction coefficient
Ext_T_High	Upper limit for the total extinction value
Ext_T_Low	Lower limit for the total extinction value
H_limit	Parameter to control the possibility that a single H-cycle shall be processed. See above (e.g. Figure 29 and Table 11)
Ext_H_onestep	Parameter to control the distribution of Ext_H during a H-cycle. Ext_H_onestep = 1 results in “step distribution” Ext_H_onestep ≠ 1 results in “triangular distribution”
Extinction_file	The name of an *Extinction file* (string) to store the extinction vector and the above parameters in this table

Table 12 Input parameters for routine “Extinction_make”

4.3.3 Output parameters

The input parameters of the routine are specified in Table 13.

Parameter name	Purpose
Ext_T_vector Output via function argument to Main Program	Vector with extinction coefficients (per simulation step)
Extinction_file	See Table 12

Table 13 Output parameters for routine “Extinction_make”

4.3.4 Examples

Table 14 gives examples of used parameters in a simulation.

Parameter name	Used value in the examples
Sim_step_sec	60
Sim_step_total	1440
Sim_step_L_My	240
Sim_step_L_Sigma	60
Sim_step_H_My	10
Sim_step_H_Sigma	4
Ext_L_My	0.4
Ext_L_Sigma	0.2
Ext_H_My	3.
Ext_H_Sigma	1.
Ext_T_High	10.
Ext_T_Low	0.32
H_limit	1.534 alt 0.674
Ext_H_onestep	1 alt 0
Extinction_file	‘Ext_1’

Table 14 Examples of used input parameters for routine “Extinction_make”

The example according to Table 14 has some alternatives regarding parameters *H_limit* and *Ext_H_onestep*. The different results are illustrated in Figure 30 (the first 2 hours), Figure 31 (the first to hours), Figure 32 (the total simulation of 24 hours) and Figure 33 (the total simulation of 24 hours).

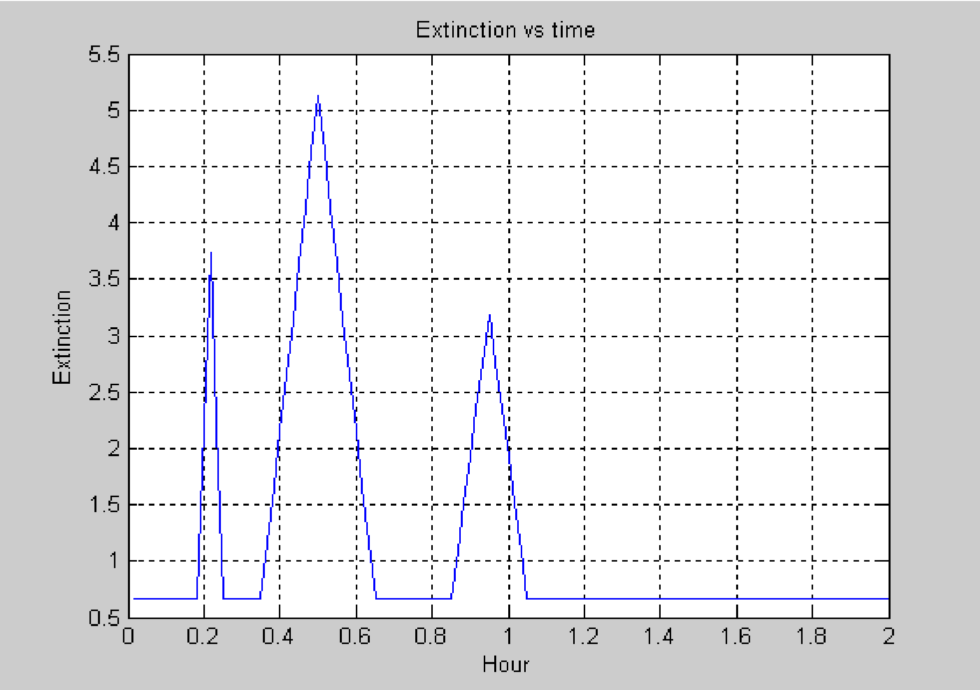

Figure 30 *H_limit* = 1.534 (corresponding to a cloudiness of 1/8)
Ext_H_onestep = 0 ("Triangular distribution" of the H-cycle)

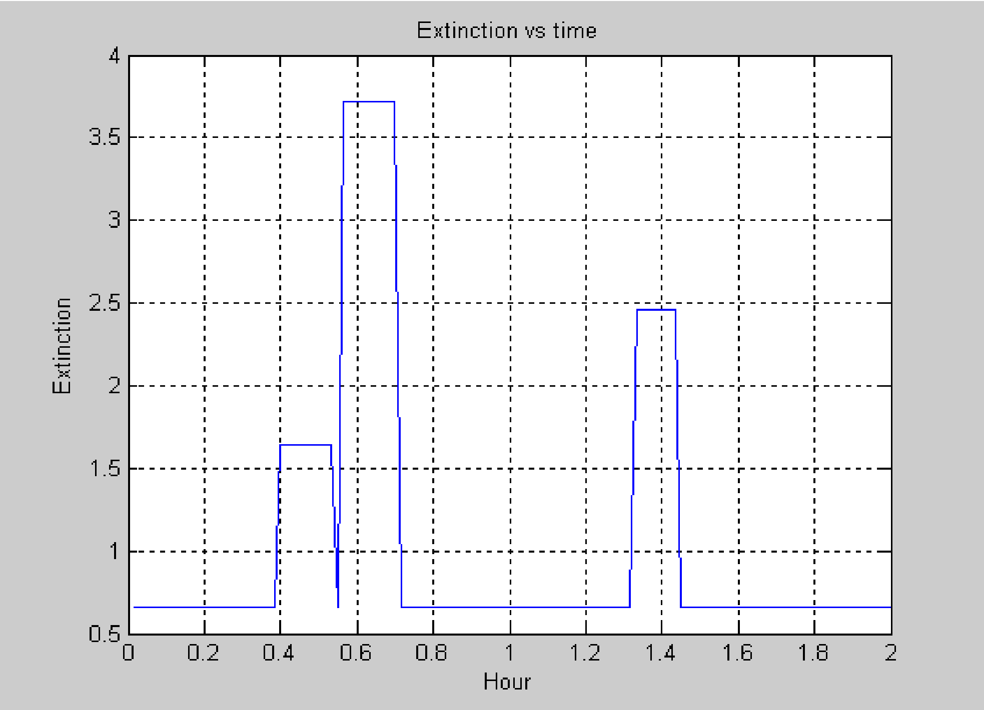

Figure 31 *H_limit* = 1.534 (corresponding to a cloudiness of 1/8)
Ext_H_onestep = 1 ("Step distribution" of the H-cycle)

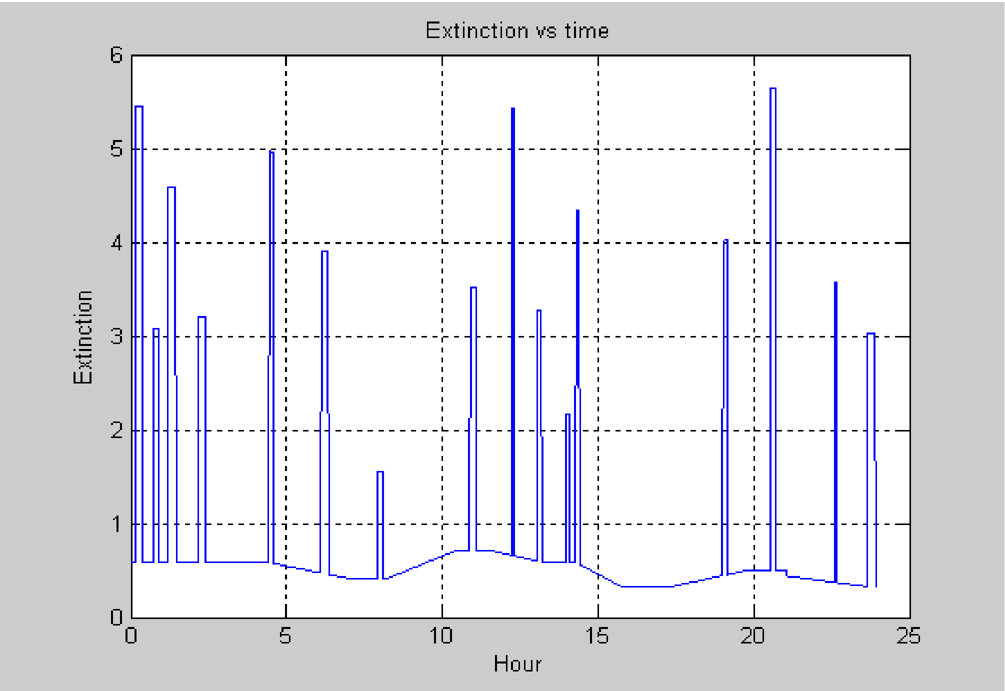

Figure 32 *H_limit* = 1.534 (corresponding to a cloudiness of 1/8)
Ext_H_onestep = 1 ("Step distribution" of the H-cycle)

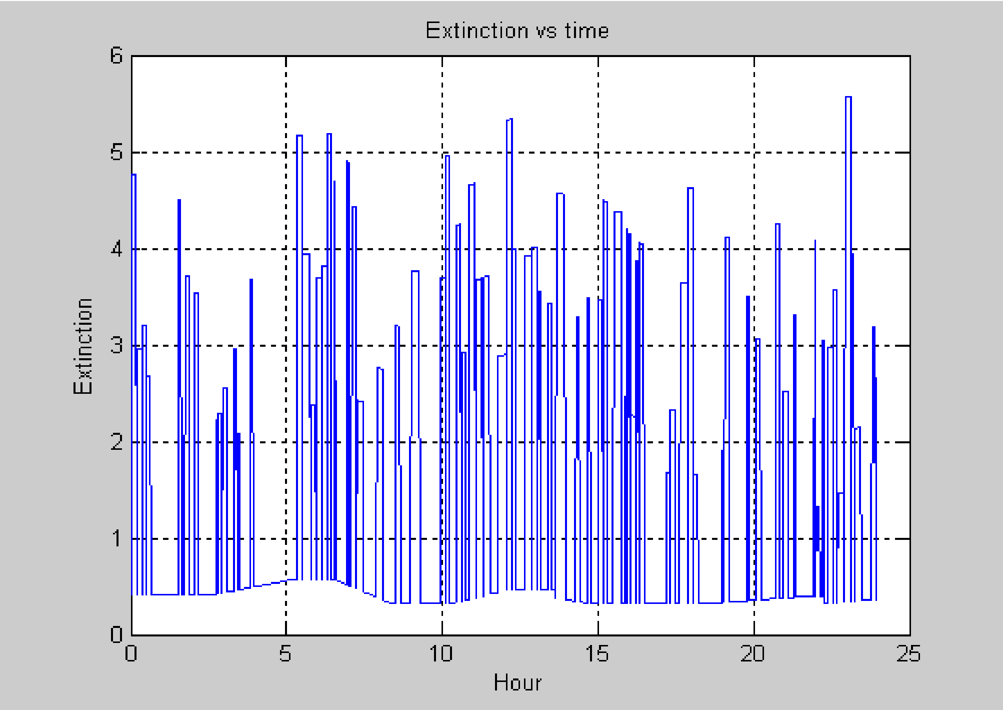

Figure 33 *H_limit* = 0.674 (corresponding to a cloudiness of 4/8)
Ext_H_onestep = 1 ("Step distribution" of the H-cycle)

4.4 Sun_intensity

4.4.1 Principle

Sun_intensity is a program function (subroutine) with the purpose to generate an irradiance vector. The term "sun irradiance" refers in this case to the sun power density (W per square meter) that is coming in to a defined surface, in the following named *Measuring surface*, at earth level, that have a specified direction to the sun. A new irradiance vector is generated for each new simulation sequence.

The irradiance values, in the following named *Radiation*, are generated as results of:

- The sun angle to zenith
- The angle relative to the sun of the surface in question
- The transmission properties of the atmosphere
- Properties regarding clear view between the sun and the surface in question

The sun position (angle to zenith and angle to south) is calculated as a result of the following:

- The local position (latitude and longitude)
- The date
- The time

The angle between the direction to the sun and the normal of the measuring surface is calculated as a function of the sun position and specified angle values regarding the surface normal to south and to zenith

The transmission properties of the atmosphere is calculated by use of the extinction vector, that is generated by the routine “Extinction_make” (see chapter 4.3).

Properties regarding clear view between the sun and the measuring surface. This depends on the sun position relative to different objects that could block the view between the sun and the surface in question. This also includes the time interval when the sun is below the horizon.

To calculate the effective irradiance against the measuring surface the following calculation sequence is used (page 43 – 52).

Equation 13

$UT = start\_hour + time\_counter / 60 \cdot count\_interval$ (in hours + decimals)

Where

start_hour :	start hour for the simulation (0 if the simulation will start at midnight)
count_interval :	simulation interval per step. This parameter is specified in minutes or parts of a minute
time_counter :	the simulation step in question. This parameter will step from 1 up to an uper limit named *time_counter_limit*

$$time\_counter\_limit = 60 \cdot (stop\_hour - start\_hour) / count\_interval$$

Where

stop_hour : stop hour for the simulation (24 if the simulation is to be performed during 24 hours, 240 if the simulation is to be performed during 10 days, and so on)

Equation 14

$$d = 367 \cdot y - floor\left(7 \cdot \left(y + floor\left((m+9)/12\right)\right)/4\right) + floor\left(275 \cdot m/9\right) + D - 730530 + UT/24$$

Where

y : the year in question (e.g. 2006)

m : the month in question (month number)

D : date (day number in the month)

This specification of *d* results in that day 0 will occur at 2000, Jan 01, 00.00 (or 1999, Dec 31, 24.00).

Note: floor(x) rounds “x” down to the nearest integer.

Equation 15

$$ecl = 23.4393 - 3.563e - 7 \cdot d$$

Where

ecl : “the obliquity of the ecliptic”, i.e. the “tilt” of the Earth’s axis of rotation (currently ca 23.4 degrees and slowly decreasing)

<u>Orbital elements of the Sun</u>

These elements is partly used in the calculation of the sun position:

N : longitude of the ascending node

i : inclination to the ecliptic (plane of the Earth’s orbit)

w : argument of perihelion

a : semi-major axis, or mean distance from Sun

e : eccentricity (0=circle, 0-1=ellipse, 1=parabola)

M : mean anomaly (0 at perihelion; increase uniformly with time)

In this case the following is on hand:

N = 0.0

i = 0.0

$w = 282.9404 + 4.70935e - 5 \cdot d$

a = 1.0

$e = 0.016709 - 1.151e - 9 \cdot d$

$M = 356.0470 + 0.9856002585 \cdot d$

<u>Compute the eccentric anomaly E from the mean anomaly M and from the eccentricity e (E and M in degrees)</u>:

Equation 16

$$E = M + e \cdot \left(\frac{180}{\pi}\right) \cdot \sin\left(\frac{M}{180}\pi\right) \cdot \left(1.0 + e \cdot \cos\left(\frac{M}{180}\pi\right)\right)$$

<u>Compute the Sun's distance r and its true anomaly v from:</u>

Equation 17

$$Xv = r \cdot \cos\left(\frac{v}{180}\pi\right) = \cos\left(\frac{E}{180}\pi\right) - e$$

Equation 18

$$Yv = r \cdot \sin\left(\frac{v}{180}\pi\right) = \sqrt{1.0 - e^2} \cdot \sin\left(\frac{E}{180}\pi\right)$$

Equation 19

$$v = a\tan 2(Yv, Xv) \cdot \frac{180}{\pi} \qquad \text{(degrees)}$$

Equation 20

$$r = \sqrt{Xv^2 + Yv^2}$$

Note: *atan2* is a function that converts a coordinate pair to the correct angle in all four quadrants.

Compute the Sun's true longitude, *lonsun*:

Equation 21

$$lonsun = v + w \qquad \text{(degrees)}$$

Convert lonsun and r to ecliptic rectangular geocentric coordinates *Xs and Ys*

Equation 22

$$Xs = r \cdot \cos\left(\frac{lonsun}{180}\pi\right)$$

Equation 23

$$Ys = r \cdot \sin\left(\frac{lonsun}{180}\pi\right)$$

As the Sun is in the ecliptic plane, *Zs* is zero. *Xs* and *Ys* is the Sun's position in a coordinate system in the plane of the ecliptic.

Convert to equatorial, rectangular, geocentric coordinates:

Equation 24

$$Xe = Xs$$

Equation 25

$$Ye = Ys \cdot \cos\left(\frac{ecl}{180}\pi\right)$$

Equation 26

$$Ze = Ys \cdot \sin\left(\frac{ecl}{180}\pi\right)$$

Compute the Sun's Right Ascension (*RA*) and Declination (*Dec*):

Equation 27

$$RA = a\tan 2(Ye, Xe) \cdot \frac{180}{\pi} \qquad \text{(degrees)}$$

Equation 28

$$Dec = a\tan 2(Ze, \sqrt{Xe^2 + Ye^2}) \cdot \frac{180}{\pi} \qquad \text{(degrees)}$$

Compute the Sun's mean longitude, *L*:

Equation 29

$$L = M + w \qquad \text{(degrees)}$$

Compute "the Sidereal Time at Greenwich", *GMST0*, at 00:00 "Universal Time":

Equation 30

$$GMST0 = L + 180 \qquad \text{(degrees)}$$

GMST0 is expressed in degrees to simplify the computations. GMST0 = 360 degrees corresponds to 24 hours, i.e. each hour corresponds to 15 degrees.

Compute "the Local Sidereal Time, *LST*:

Equation 31

$$LST = GMST0 + UT \cdot 15.0 + long \qquad \text{(degrees)}$$

Where

UT : "Universal Time" expressed in hours + decimals

$Long$: local longitude in degrees. East longitude counts as positive and west longitude as negative

Compute "the Sun's Local Hour Angle, LHA, i.e. the angle the Earth has turned since the Sun last was in the south:

Equation 32

$$LHA = LST - RA$$

<u>Compute the Sun's altitude above the horizon, *alpha*:</u>

Equation 33

$$sin\_alpha = \cos\left(\frac{Dec}{180}\pi\right)\cdot\cos\left(\frac{lat}{180}\pi\right)\cdot\cos\left(\frac{LHA}{180}\pi\right)+\sin\left(\frac{Dec}{180}\pi\right)\cdot\sin\left(\frac{lat}{180}\pi\right)$$

Where

lat : the latitude in question

Equation 34

$$alpha = \arcsin(sin\_alpha) \qquad \text{(radians)}$$

<u>Compute the Sun's azimuth, *az*</u>

Equation 35

$$cos\_az = \frac{\cos\left(\frac{Dec}{180}\pi\right)\cdot\sin\left(\frac{lat}{180}\pi\right)\cdot\cos\left(\frac{LHA}{180}\pi\right)-\sin\left(\frac{Dec}{180}\pi\right)\cdot\cos\left(\frac{lat}{180}\pi\right)}{\cos(alpha)}$$

Equation 36

$$az = \arccos(cos\_az)) \qquad \text{(radians)}$$

<u>Compute the "atmospheric depth" as a function of the Sun's altitude above the horizon, *alpha*:</u>

Equation 37

$$M\_atm = \frac{1}{\sin(alpha)}$$

Where

M_atm : the atmospheric depth relative to the depth when the Sun is in zenith ($alpha = \frac{\pi}{2}$)

Compute the atmospheric transmission, τ :

Equation 38

$$\tau = \exp\left(-Extinction \cdot M\_atm\right)$$

Where

Extinction : extinction coefficient for the atmospheric depth when the Sun is in zenith

Compute the Sun irradiation that is incoming to the measuring surface:

Equation 39

$$Radiation\_A = \tau \cdot Radiation\_ref$$

Where

Radiation_ref : Sun irradiation before passing the atmosphere

Radiation_A : Sun irradiation after passing the atmosphere

<u>Compute the angle, *beta*, between the direction to the Sun and the measuring surface normal (the surface that corresponds to the solar cell panel)</u>:

Equation 40

$$cos\_beta = \sin(alpha)\cdot\cos\left(\frac{Srf\_rel\_Z}{180}\pi\right)+\cos(alpha)\cdot\sin\left(\frac{Srf\_rel\_Z}{180}\pi\right)\cdot\cos\left(az-\frac{Srf\_rel\_S}{180}\pi\right)$$

Where

Srf_rel_Z : the normal angle of the measuring surface relative to zenith

Srf_rel_S : the normal angle of the measuring surface relative to south

Equation 41

$$beta = \arccos(cos\_beta) \qquad \text{(radians)}$$

<u>Compute the ”effective irradiation” from the Sun against the measuring surface as a function of the angle *beta*:</u>

Equation 42

$$Radiation\_B = Radiation\_A\cdot cos\_beta$$

Where

Radiation_A : to the surface incoming irradiation

Radiation_B : effective part of irradiation

Function for “Sun Tracking”

There is a function in the routine that simulates so called “Sun Tracking”. This means that the measuring surface follows the Sun position, i.e. the angle, *beta*, is assigned the value zero. To get “Sun Tracking” activated the parameter “Tracking” should be assigned the value 1.

Properties regarding clear view between the sun and the measuring surface

If there is no clear view between the sun and the measuring surface the calculated irradiance is assigned the value zero. Two equations should be fulfilled for clear view:

Equation 43 : $alpha \geq alpha\_\min$

Equation 44 : $azimuth\_\min \leq az \leq azimuth\_\max$

Where

alpha : Sun’s altitude above the horizon

alpha_min : under limit for the altitude of the Sun to be visible

azimuth_min : under limit for the azimuth of the Sun to be visible

azimuth_max : upper limit for the azimuth of the Sun to be visible

4.4.2 Input parameters

The input parameters of the routine are specified in Table 15.

Parameter name	**Purpose**
Sim_step_sec Input via function argument from Main Program	Time interval in seconds per simulation step (60 is a standard value)
Sim_step_total Input via function argument from Main Program	The total number of simulation steps per sequence (Sim_step_total = 43200 corresponds to a simulation sequence over a time of 30 days if Sim_step_sec = 60)
Ext_T_vector Input via function argument from Main Program	Vector with extinction coefficients (per simulation step)
y	Start year for simulation
m	Start month for simulation
D	Start date for simulation
long	Longitude
lat	Latitude
start_hour	Start hour for simulation
Surface_rel_S	The normal of the measuring surface relative to south (degrees). If no “tracking”
Surface_rel_Z	The normal of the measuring surface relative to zenith (degrees). If no “tracking”
Radiation_ref	The sun irradiance before passing the atmosphere (W / m^2)
azimuth_min	Under limit for the azimuth of the Sun to be visible
azimuth_max	Upper limit for the azimuth of the Sun to be visible
alpha_min	Under limit for the altitude of the Sun to be visible
Tracking	If “Tracking” = 1 then the measuring surface follows the Sun position (“sun tracking”)

Table 15 Input parameters for routine “Sun_intensity”

4.4.3 Output parameters

The output parameters of the routine are specified in Table 16.

Parameter name	Purpose
Radiation_vector Output via function argument to Main Program	Vector with sun irradiation (effective to the solar cells) (per simulation step)

Table 16 Output parameters for routine “Sun_intensity”

4.4.4 Examples

Table 17 gives examples of used parameters in a simulation.

Parameter name	Used value in the examples
Sim_step_sec Input via function argument from Main Program	60
Sim_step_total Input via function argument from Main Program	1440 (corresponds to 24 hours simulation per sequence) 525600 (corresponds to 365 days simulation per sequence)
y	2006
m	3, 6, 12
D	21
long	11.97, 20.22, 36.83
lat	57.72, 67.85, -1.28
start_hour	0
Surface_rel_S	0.
Surface_rel_Z	45.

Radiation_ref	1367.
azimuth_min	-90.
azimuth_max	90.
alpha_min	0.
Tracking	0, 1

Table 17 Examples of used input parameters for routine “Sun_intensity”

4.4.4.1 Different localities and seasons

Table 18 and Table 19 specifies three localities respectively three seasons that have been compared regarding different conditions for Sun irradiation. Some results follow in Figure 34, Figure 35 and Figure 36 and Table 20, Table 21 and Table 22. In the figures follow the Sun irradiation as a function of time during the day. The tables give the integrated irradiation during the day. In these examples the extinction coefficient consequently has been assumed to 0.3126. This extinction value results in an irradiance at sea level of 1000 W/m^2 , if the Sun is at zenith (Radiation_ref = 1367 according to Table 17).

Location	Latitude (degrees)	Longitude (degrees)
Nairobi	-1.283	36.833
Kiruna	67.850	20.217
Göteborg	57.710	11.968

Table 18 The different places that have been studied regarding Sun irradiation

Note! To get a more clear curve illustration, the time scale in the figures have been adapted to local time in Göteborg.

Season	Month	Date
Summer	June	20
Autumn	September	20
Winter	December	20

Table 19 The different seasons that have been studied regarding Sun irradiation

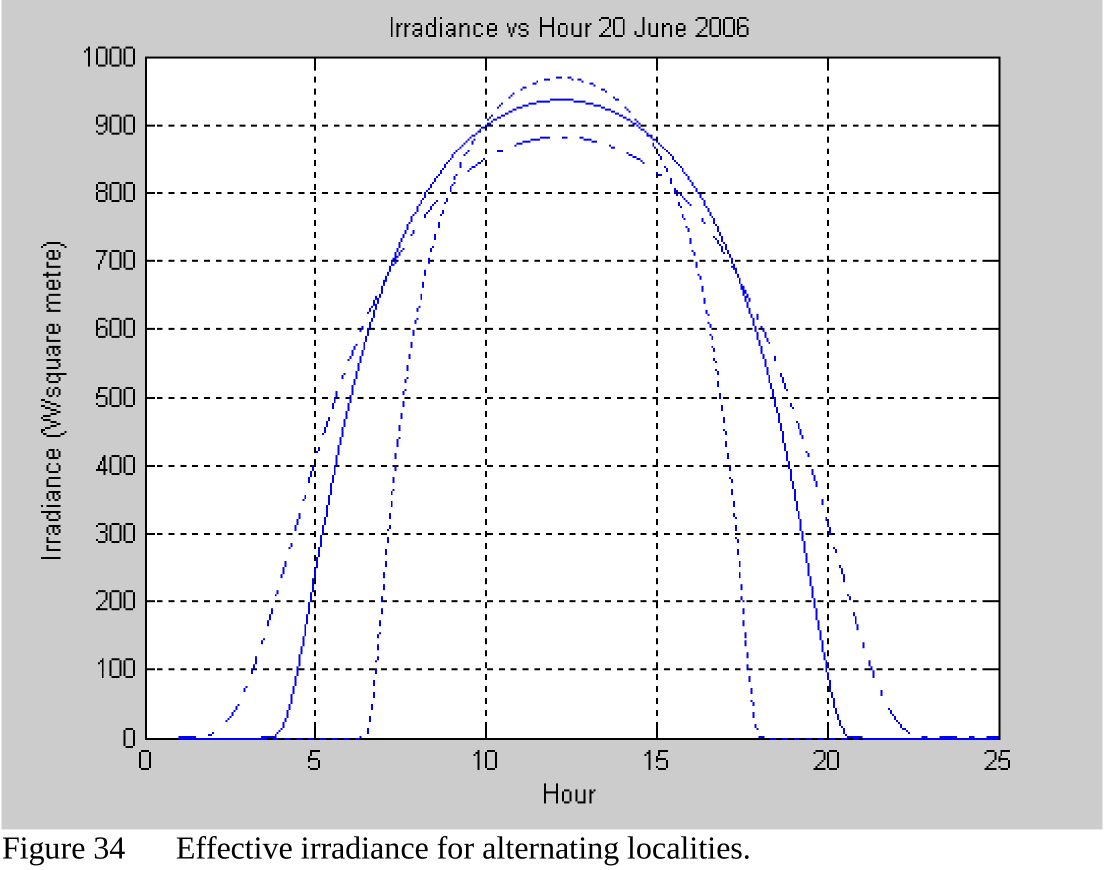

Figure 34 Effective irradiance for alternating localities.
Date: 20 June 2006. See Table 20

Curve identifier (Figure 34)	Locality	Integration (kWh / m^2)
'…….'	Nairobi	8.42
'-.-.-.-'	Kiruna	11.77
'______'	Göteborg	11.11

Table 20 Curve identifier and integrated irradiances. 20 June

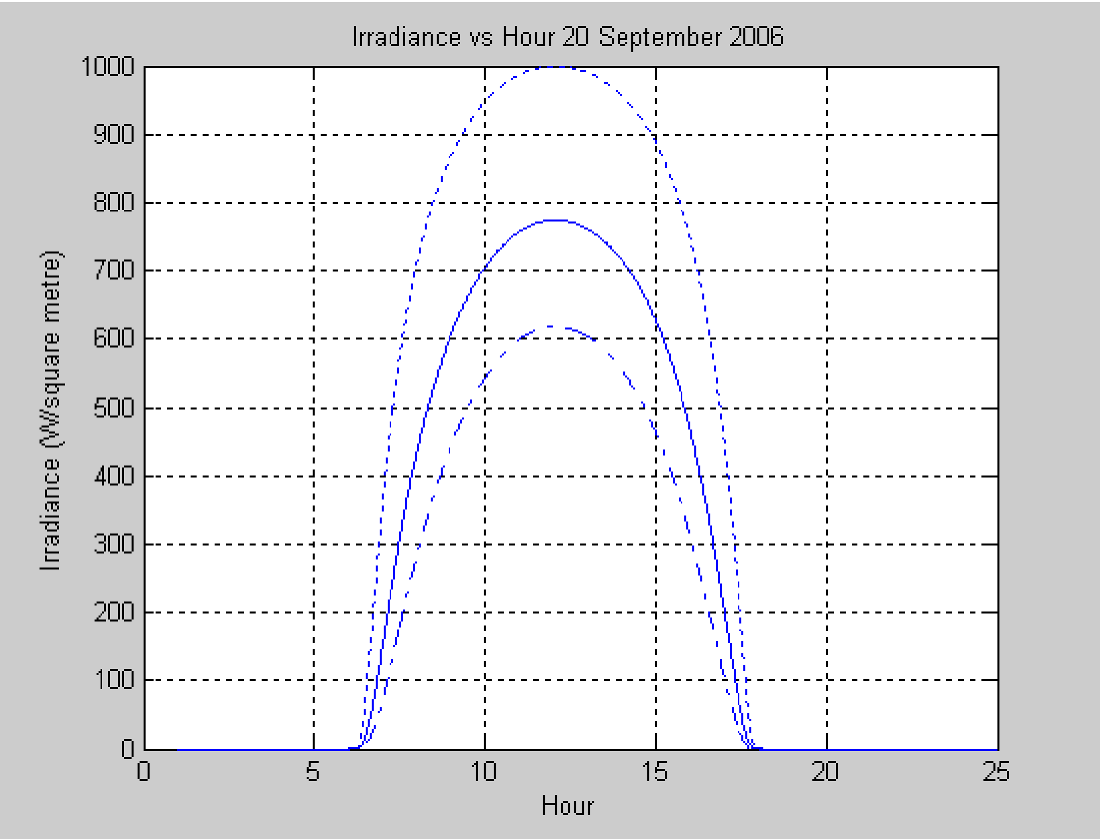

Figure 35 Effective irradiance for alternating localities. See Table 21

Curve identifier (Figure 35)	Locality	Integration (kWh / m^2)
'…….'	Nairobi	8.89
'-.-.-.-'	Kiruna	4.60
'______'	Göteborg	6.20

Table 21 Curve identifier and integrated irradiances. 20 September

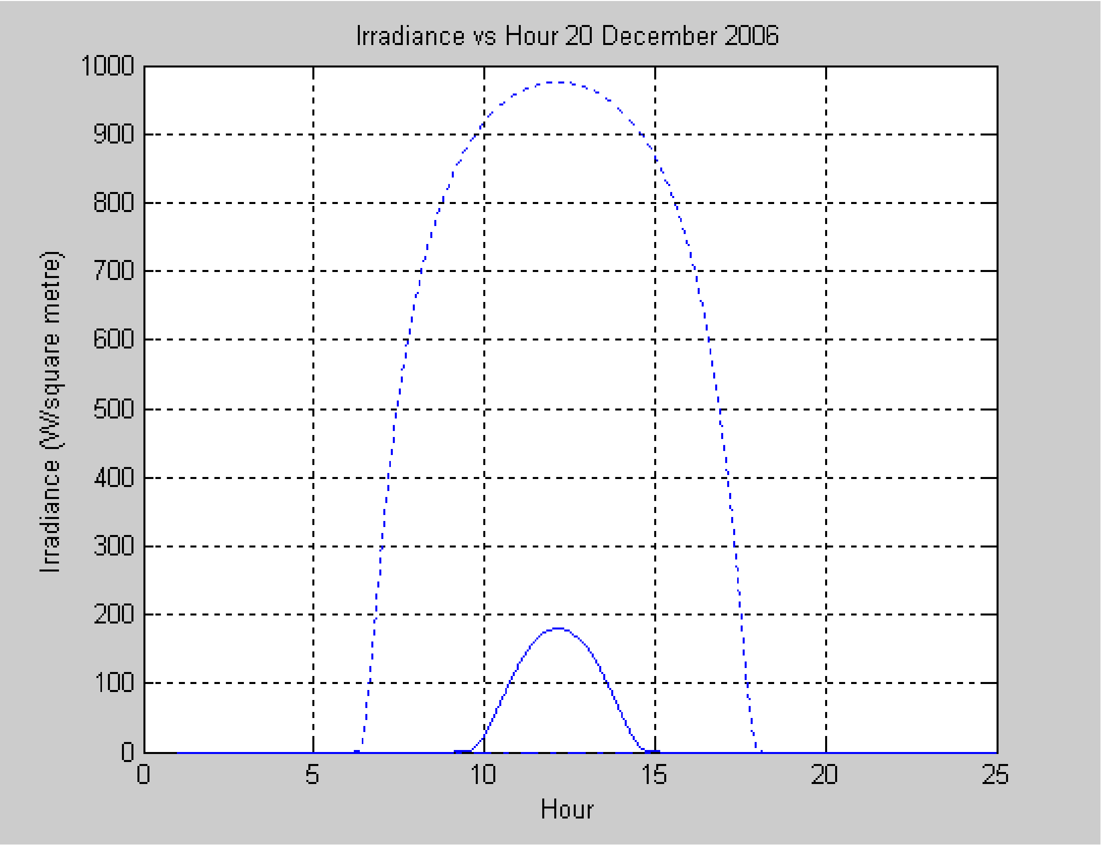

Figure 36 Effective irradiance for alternating localities. See Table 22

Curve identifier (Figure 36)	Locality	Integration (kWh / m^2)
'…….'	Nairobi	8.61
'-.-.-.-'	Kiruna	0
'______'	Göteborg	0.54

Table 22 Curve identifier and integrated irradiance. 20 December. Note! Sun not over the horizon in Kiruna

4.4.4.2 Integrating irradiances per month and summed over 12 month

For calculated integrated irradiances See Table 23 and Figure 37.

Month	Integrated irradiance (kWh/m^2) Nairobi	Integrated irradiance (kWh/m^2) Göteborg	Integrated irradiance (kWh/m^2) Kiruna
January	260.36	32.06	0.16
February	264.96	91.08	27.43
Mars	266.68	165.52	111.65
April	263.51	244.25	218.98
May	257.33	302.48	304.27
June	253.12	330.93	348.99
July	255.08	318.29	328.74
Augusty	261.27	269.22	254.84
September	266.08	197.15	153.59
Oktober	266.05	117.67	53.70

November	261.92	47.57	2.42
December	258.61	18.41	0
Total year	3135	2135	1805

Table 23 Integrated irradiance per month and summed over 12month

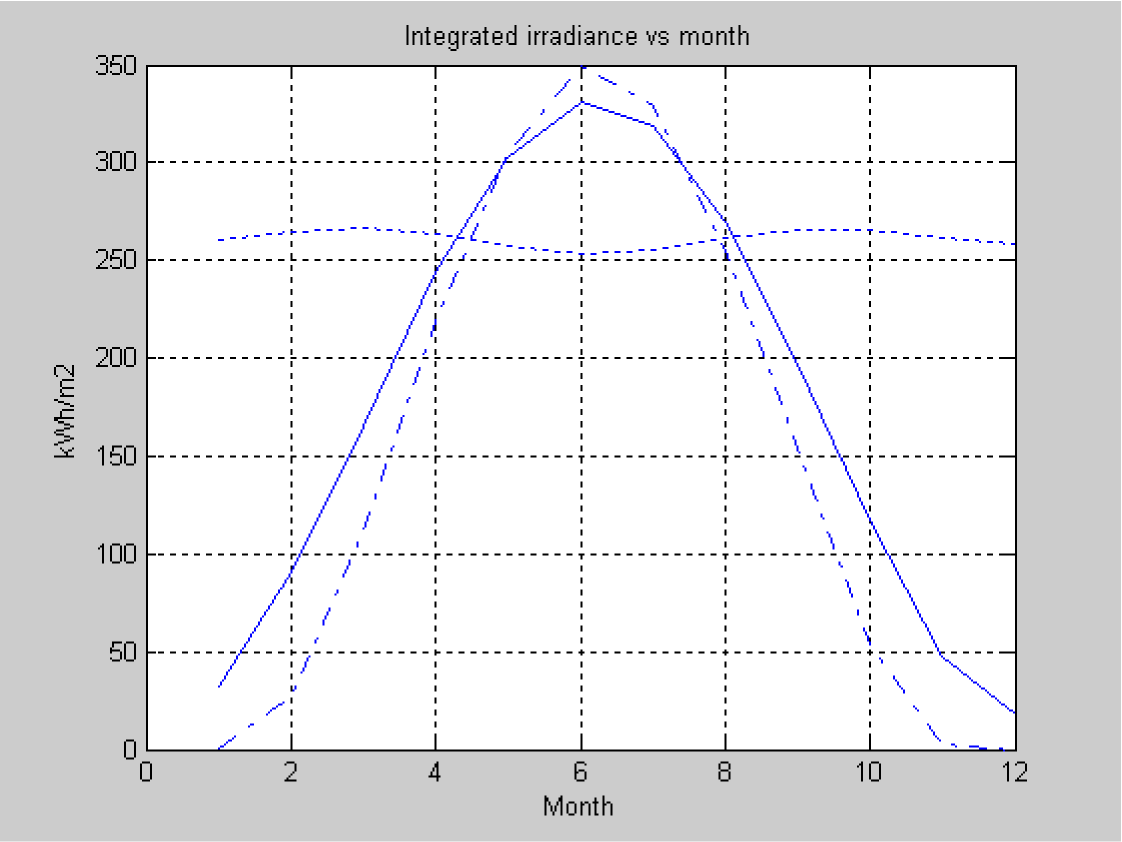

Figure 37 Integrated irradiance vs month

'.......' Nairobi

'______' Göteborg

'-.-.-.-' Kiruna

4.4.4.3 Toplevels of irradiances over a year

Calculated toplevels of irradiances over a year for Göteborg, Kiruna and Nairobi are illustrated in Figure 38, Figure 39 and Figure 40.

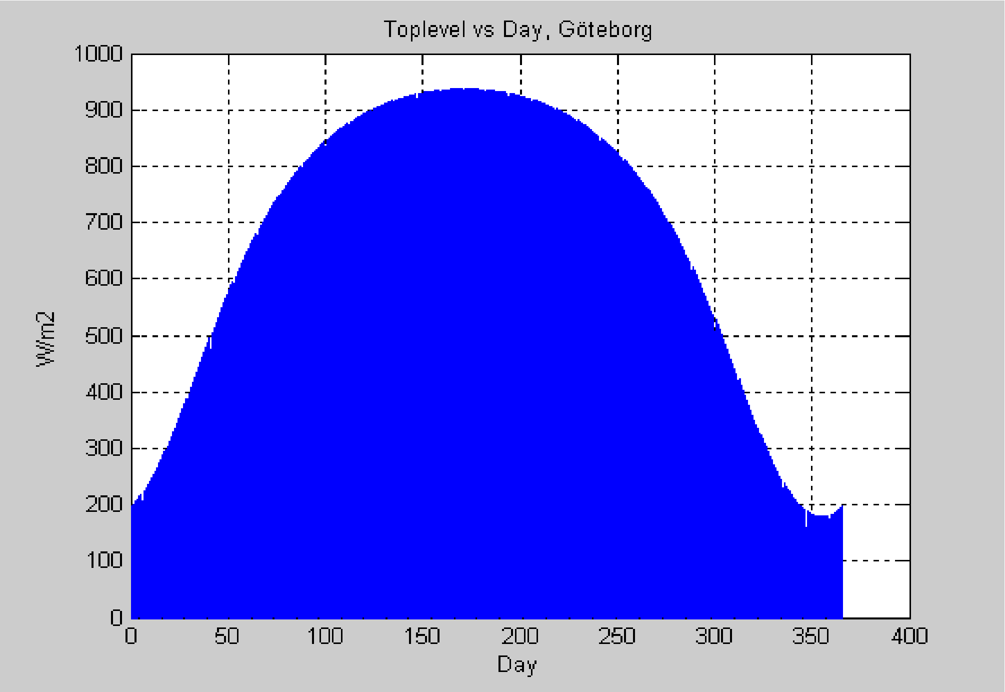

Figure 38 Top level irradiance vs day over a year. Göteborg

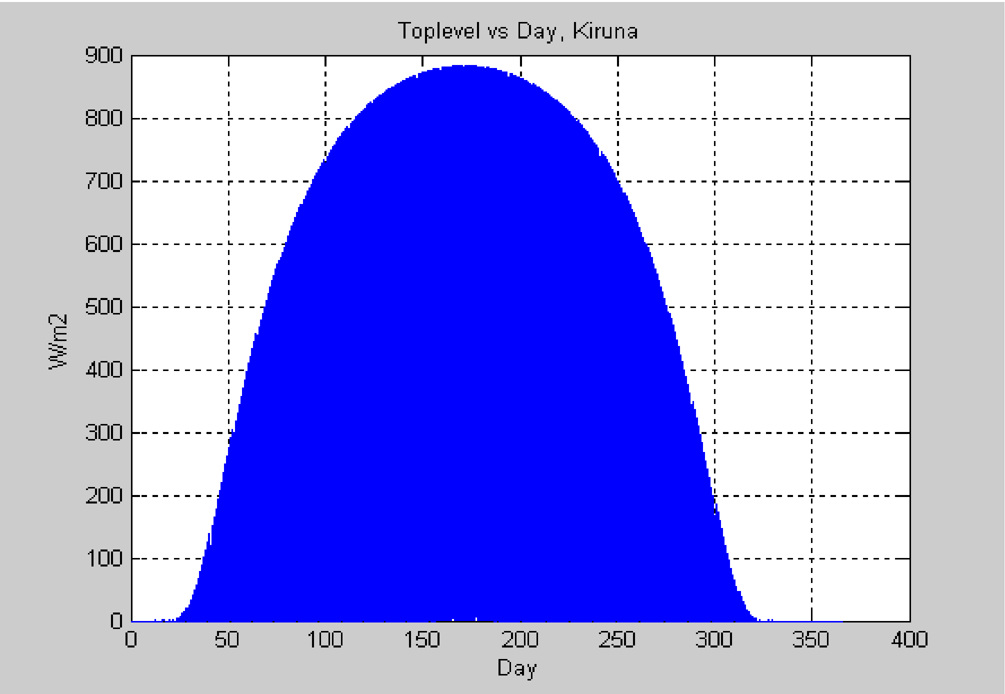

Figure 39 Top level irradiance vs day over a year. Kiruna

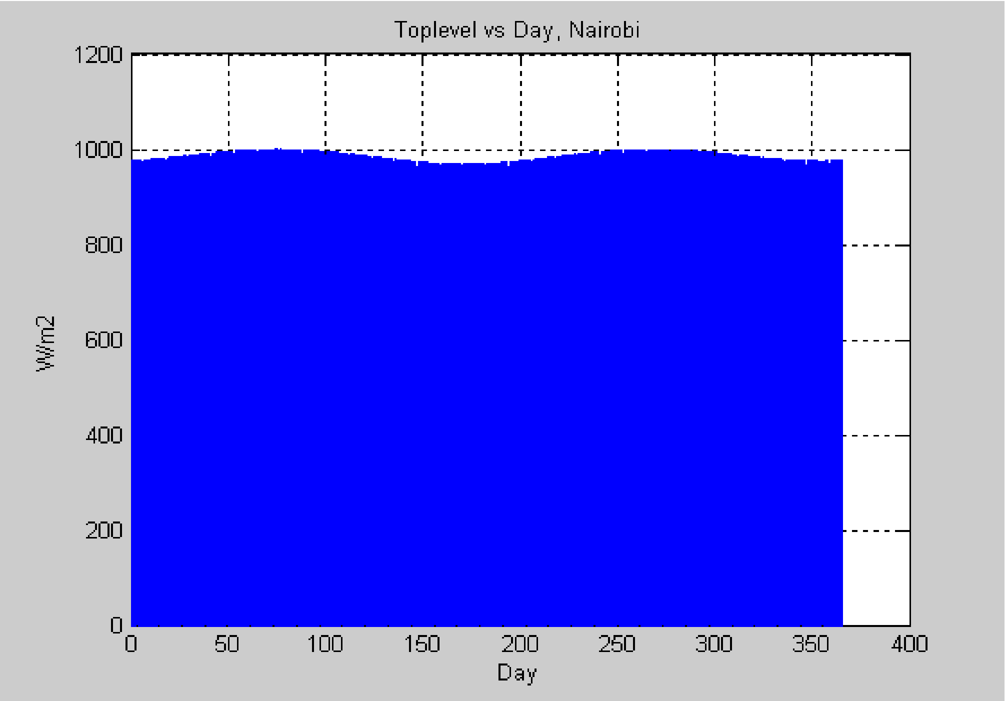

Figure 40 Top level irradiance vs day over a year. Nairobi

4.4.4.4 Different angles between measuring surface and zenit respectively "Sun Tracking"

Figure 41 (June) and Figure 43 (September) illustrate the dependence on the effective irradiance when the measuring surface relative zenith alternates. The angle between the measuring surface (normal) and South has been assigned to zero. Figure 42 (June) and Figure 44 (September) show the corresponding results if "Sun Tracking" is used. The integrated irradiances follow in Table 24 (June) and Table 25 (September).

The locality in these calculations has been Göteborg.

In could bee interesting to compare the results between *Surface_rel_Z* = 0 respectively *Surface_rel_Z* = 90 from the two sesons. In June *Surface_rel_Z* = 0 is much to prefere before *Surface_rel_Z* = 90 while in September *Surface_rel_Z* = 90 is much to prefere before *Surface_rel_Z* = 0. This is of course an effect of the lower Sun altitude over the horizon in September compared with in June.

An imported point to observe is the great power increase that is on hand by using "Sun Tracking".

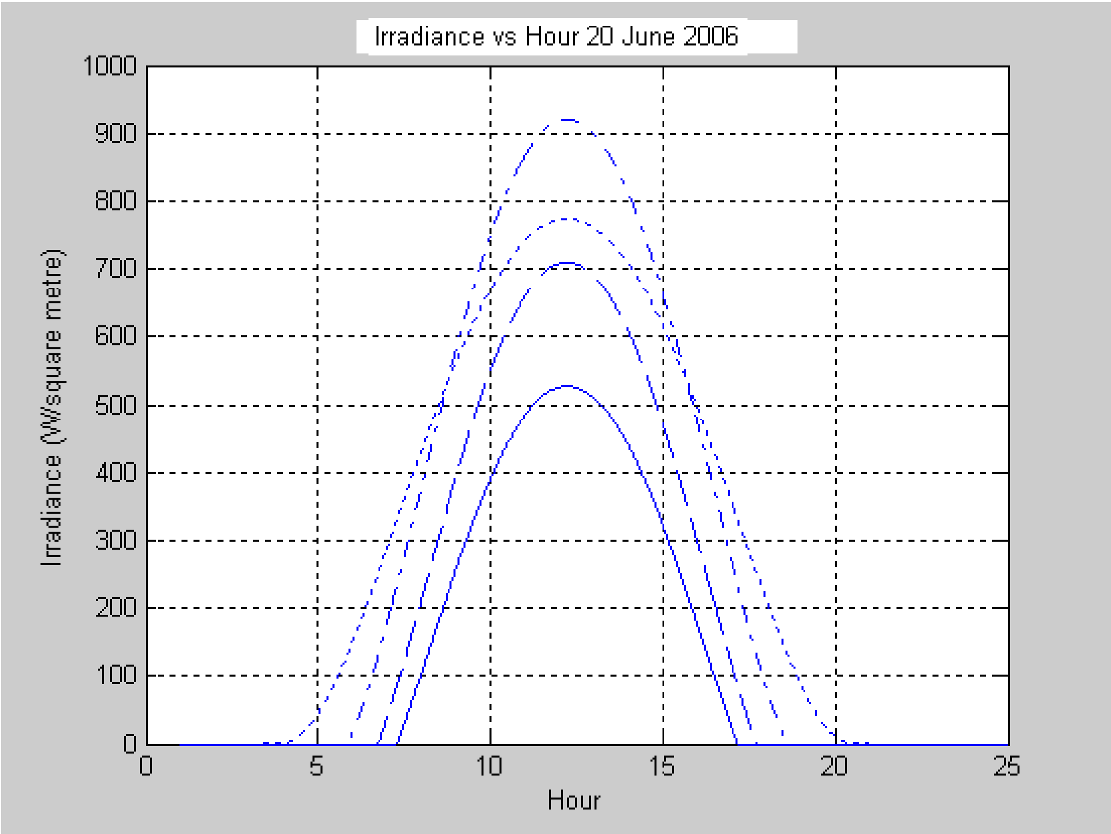

Figure 41 Effective irradiance for alternating angle of the measuring surface relative zenith. 20 June 2006. Göteborg. See Table 24

Curve identifier (Figure 41)	*Surface_rel_Z*	Integration (kWh / m^2)
'…….'	0	6.91
'-.-.-.-'	45	6.89
'__ __'	75	4.74
'______'	90	3.23
See Figure 42	"Sun Tracking"	11.11

Table 24 Integrated irradiance for different surface angles rel zenith. 20 June 2006. Göteborg

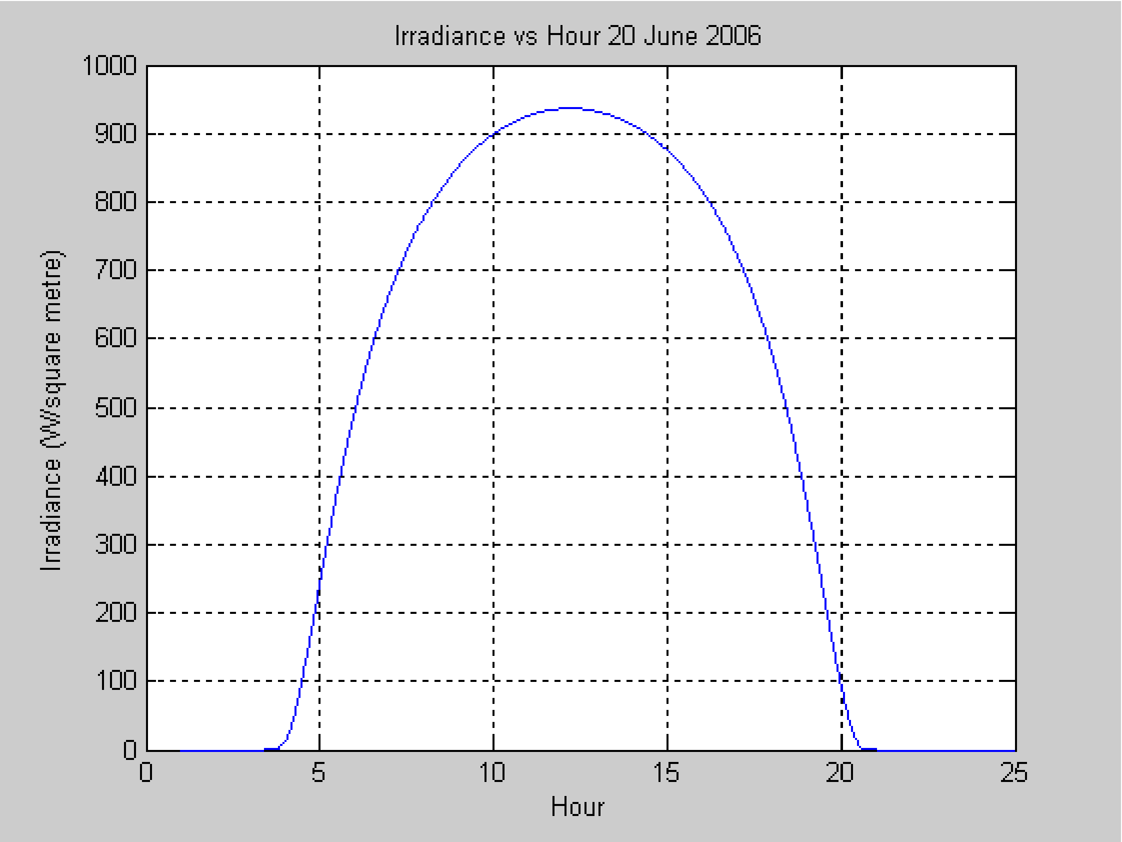

Figure 42 Effective irradiance if “Sun Tracking” is used. 20 June 2006. Göteborg

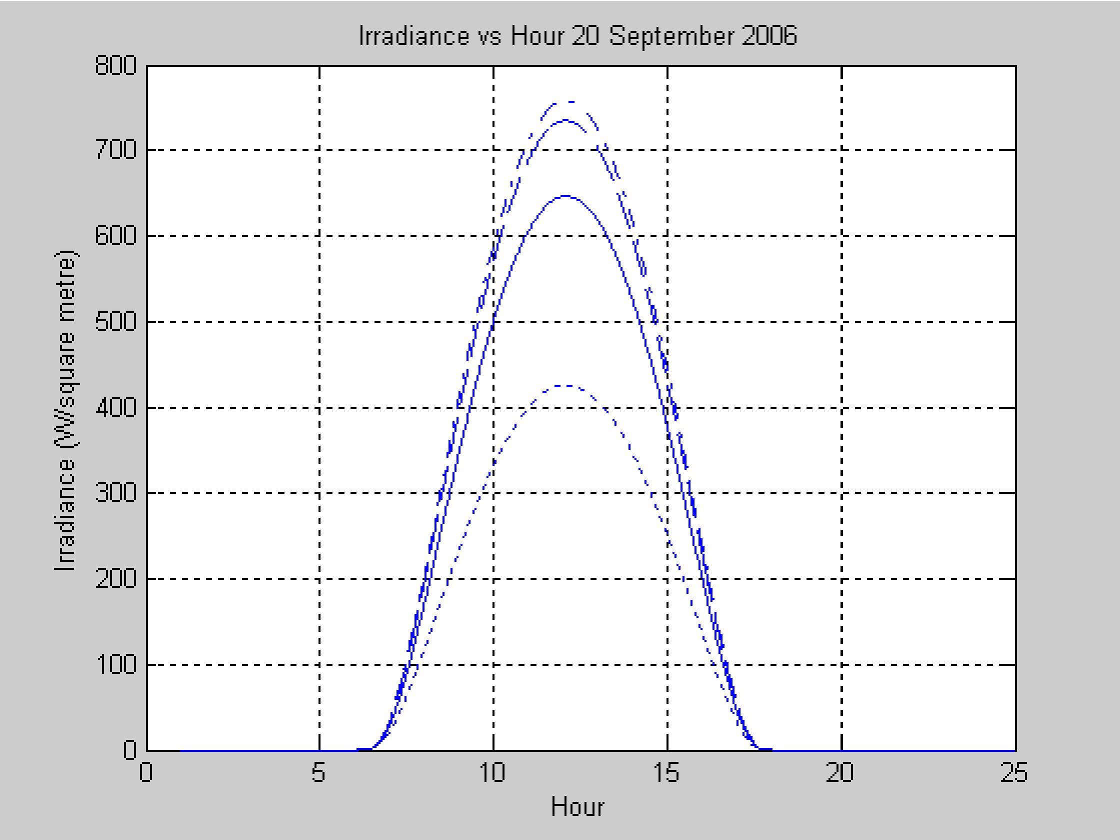

Figure 43 Effective irradiance for alternating angle of the measuring surface relative zenith. 20 September 2006. Göteborg. See Table 25

Curve identifier (Figure 43)	*Surface_rel_Z*	Integration (kWh / m^2)
'…….'	0	2.71
'-.-.-.-'	45	4.79
'— —'	75	4.63
'______'	90	4.07
See Figure 44	“Sun Tracking”	6.20

Table 25 Integrated irradiance for different surface angles rel zenith. 20 September 2006. Göteborg

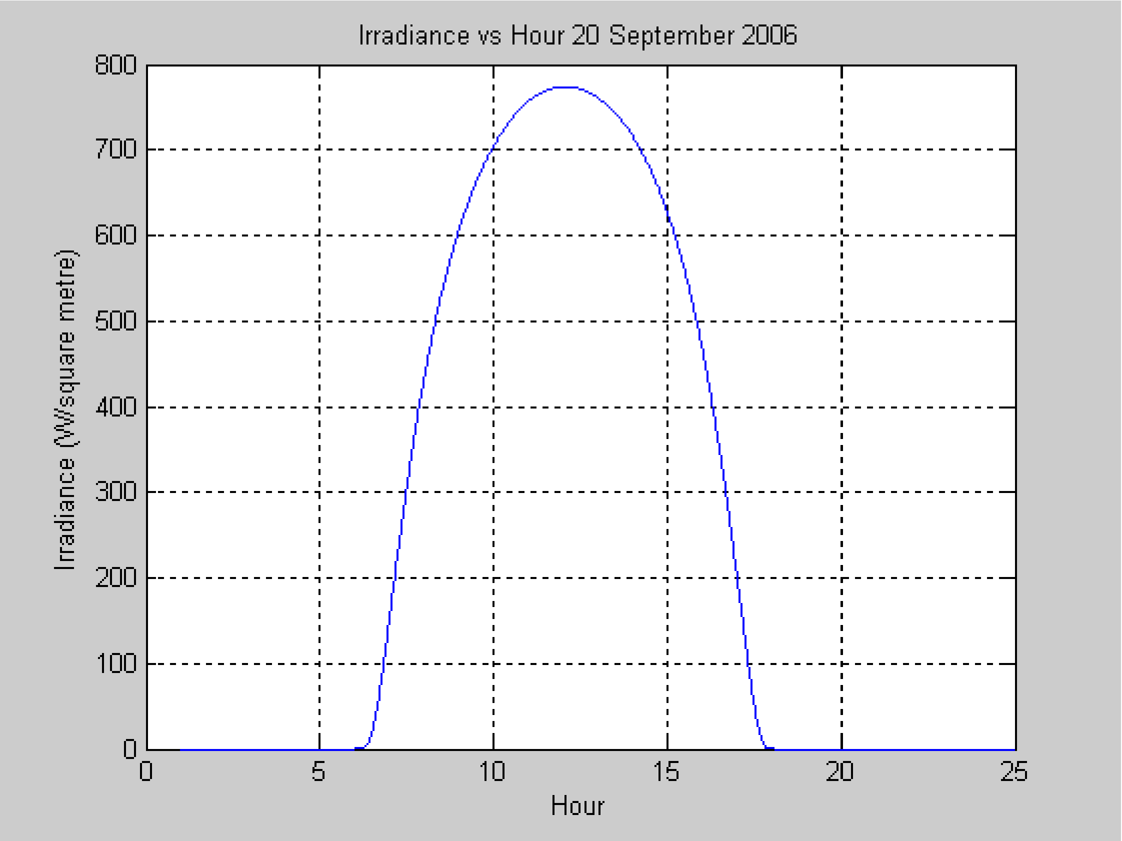

Figure 44 Effective irradiance if “Sun Tracking” is used. 20 September 2006. Göteborg

Table 26, Table 27 and Table 28 show the integrated irradiance during a year for Göteborg, Kiruna and Nairobi with alternating values regarding *Surface_rel_Z* (SRZ) and *Surface_rel_*S (SRS). The tables refere to figures showing the toplevels during a year and also how the integrated irradiance will vary as a function of SRZ. For Göteborg and Kiruna the SRS - value consequently has been assigned to 0. For the Nairobi examples the SRS has been assigned to both 0 and 180. As a comparison the integrated irradiance for "Sun Tracking" has been noted in the tables.

Surface_rel_Z	Integration (kWh / m^2)
0	1116 (also see Figure 45)
15	1342 (also see Figure 46)
30	1479 (also see Figure 47)
40	1516 (also see Figure 48)
41	1518
42	1518
43	1519
44	1519
45	1518 (also see Figure 49)
46	1517
50	1509 (also see Figure 50)
60	1458 (also see Figure 51)
75	1302 (also see Figure 52)
90	1062 (also see Figure 53)
"Sun Tracking"	2167

Table 26 Integrated irradiance during a year for different zenith angles and "Sun Tracking". SRS = 0. Göteborg

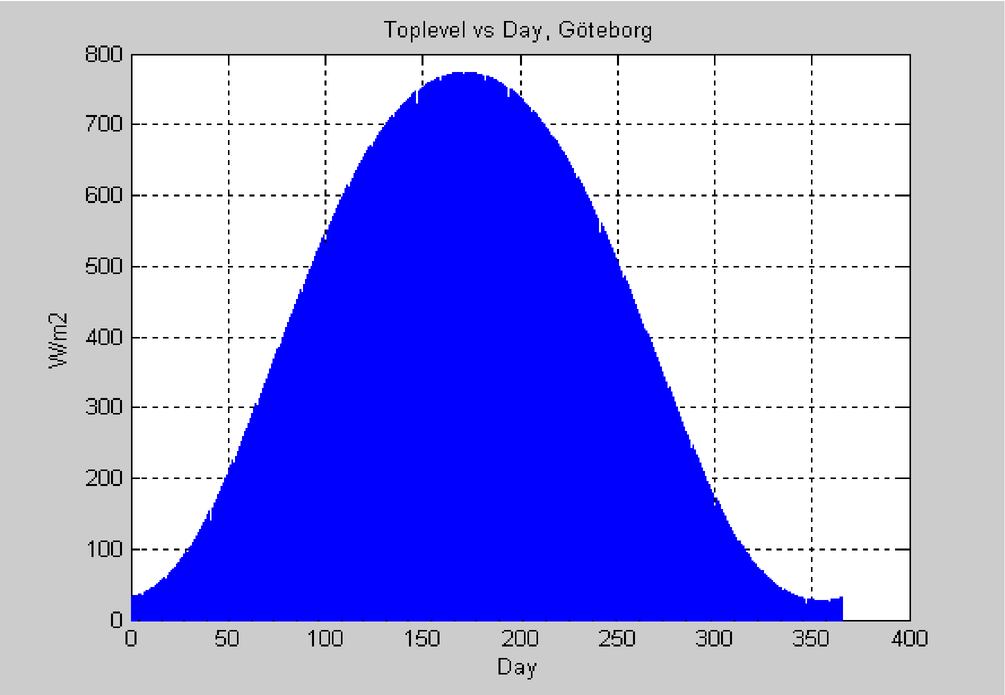

Figure 45 SRZ = 0 and SRS = 0. Toplevel vs day during a year. Göteborg

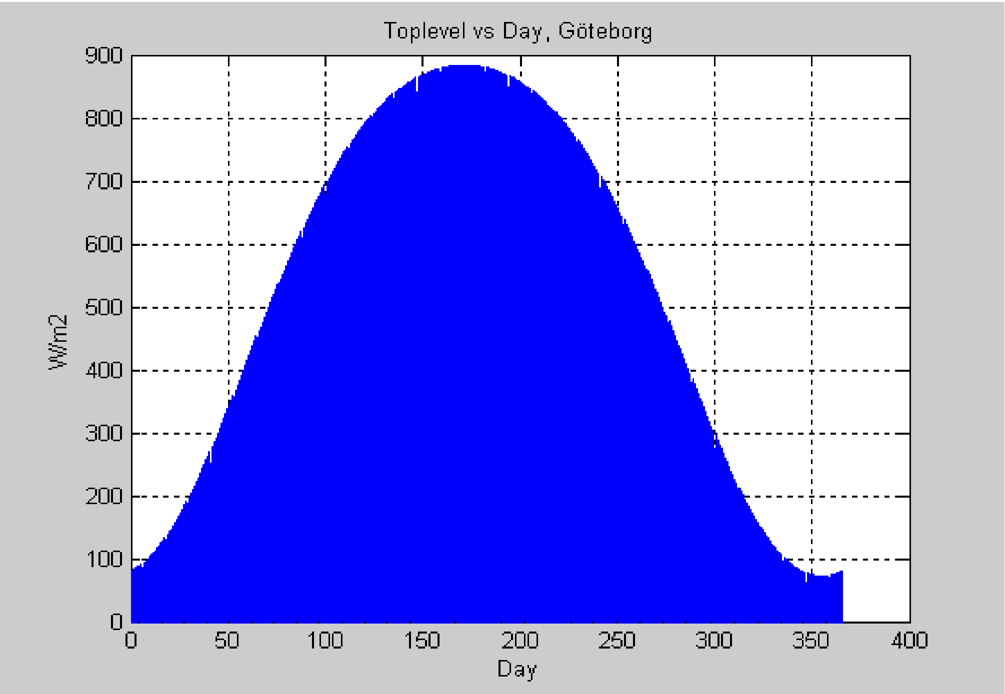

Figure 46 SRZ = 15 and SRS = 0. Toplevel vs day during a year. Göteborg

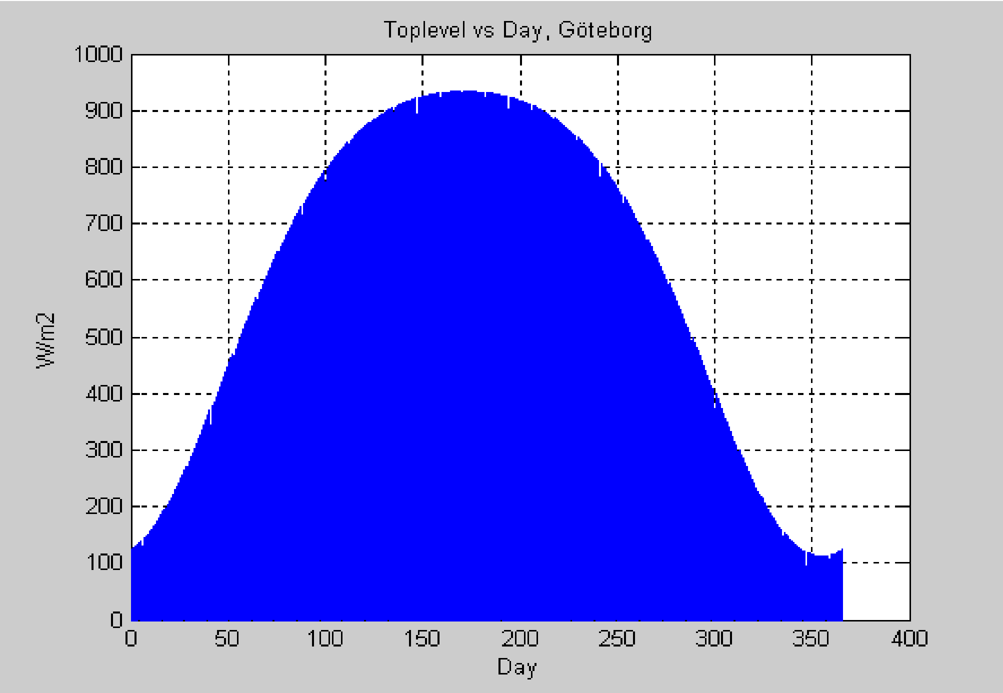

Figure 47 SRZ = 30 and SRS = 0. Toplevel vs day during a year. Göteborg

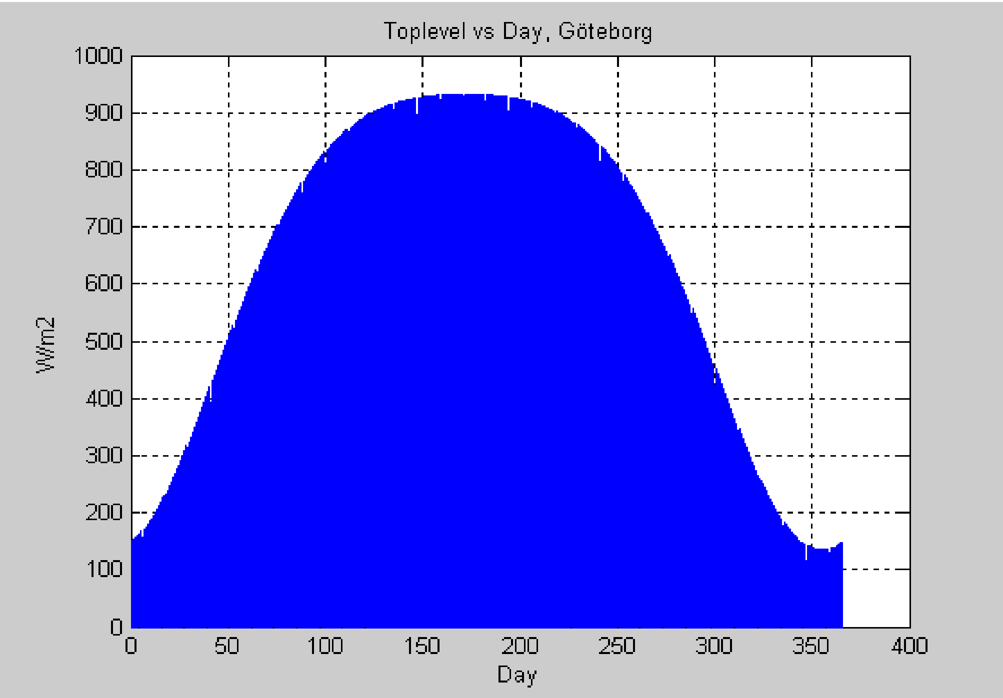

Figure 48 SRZ = 40 and SRS = 0. Toplevel vs day during a year. Göteborg

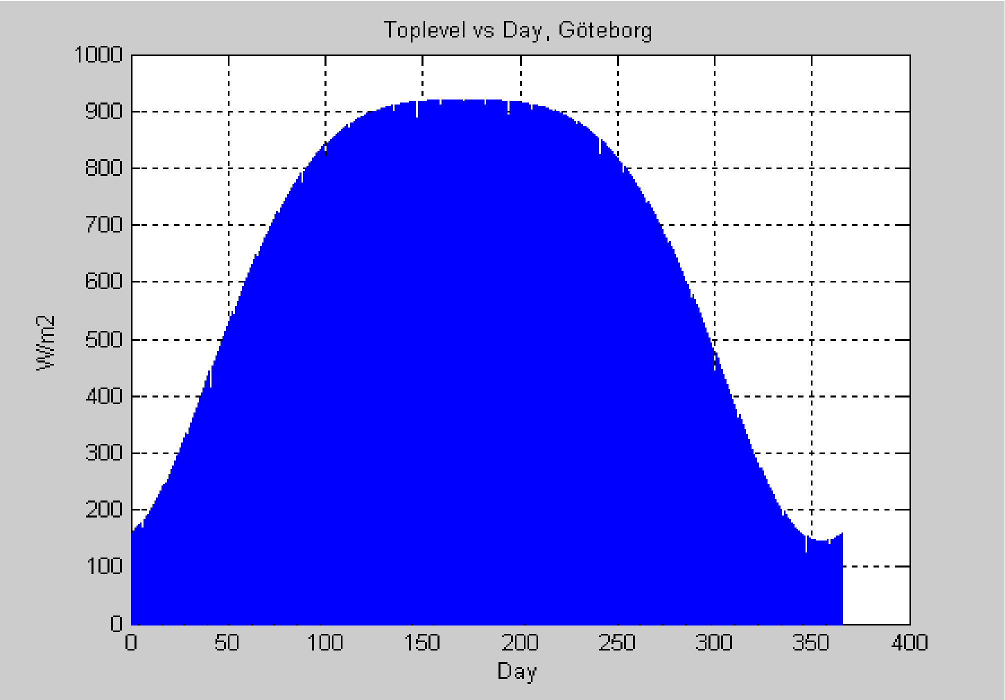

Figure 49 SRZ = 45 and SRS = 0. Toplevel vs day during a year

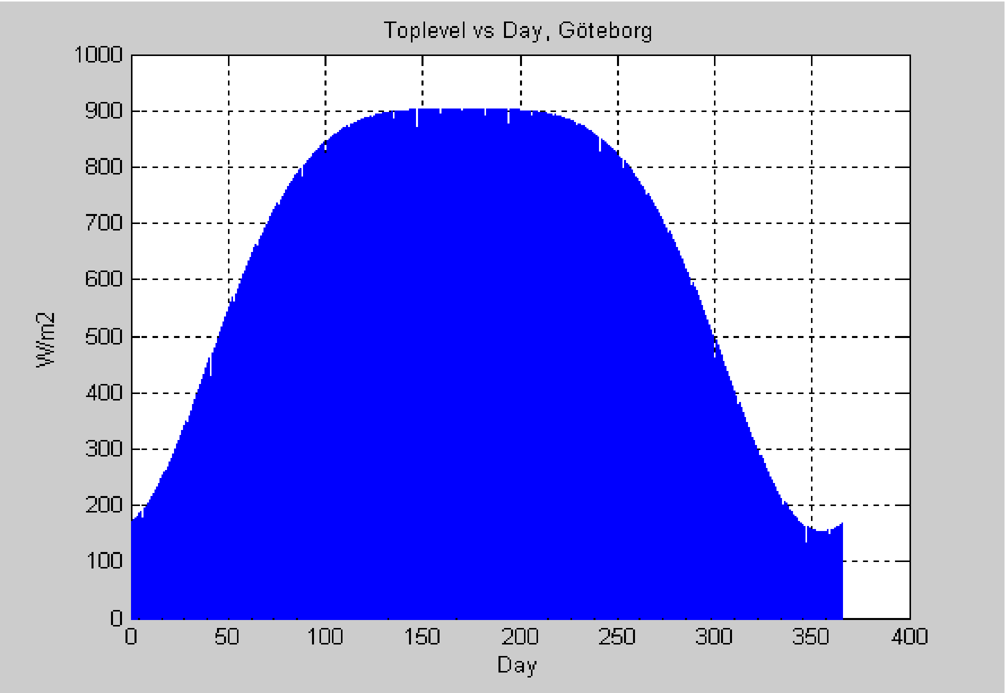

Figure 50 SRZ = 50 and SRS = 0. Toplevel vs day during a year. Göteborg

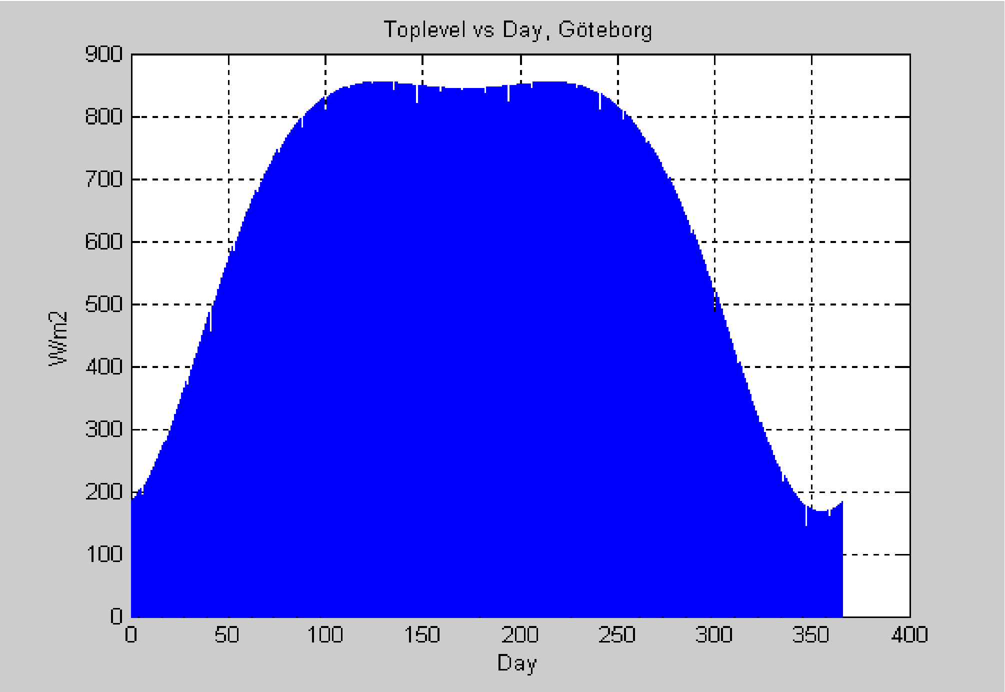

Figure 51 SRZ = 60 and SRS = 0. Toplevel vs day during a year. Göteborg

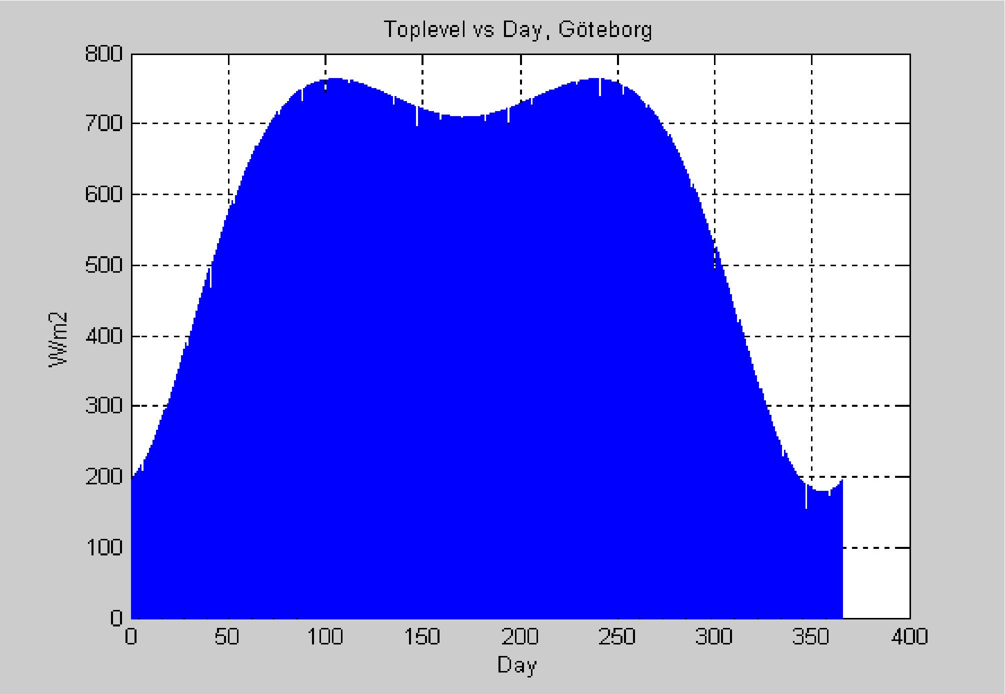

Figure 52 SRZ = 75 and SRS = 0. Toplevel vs day during a year. Göteborg

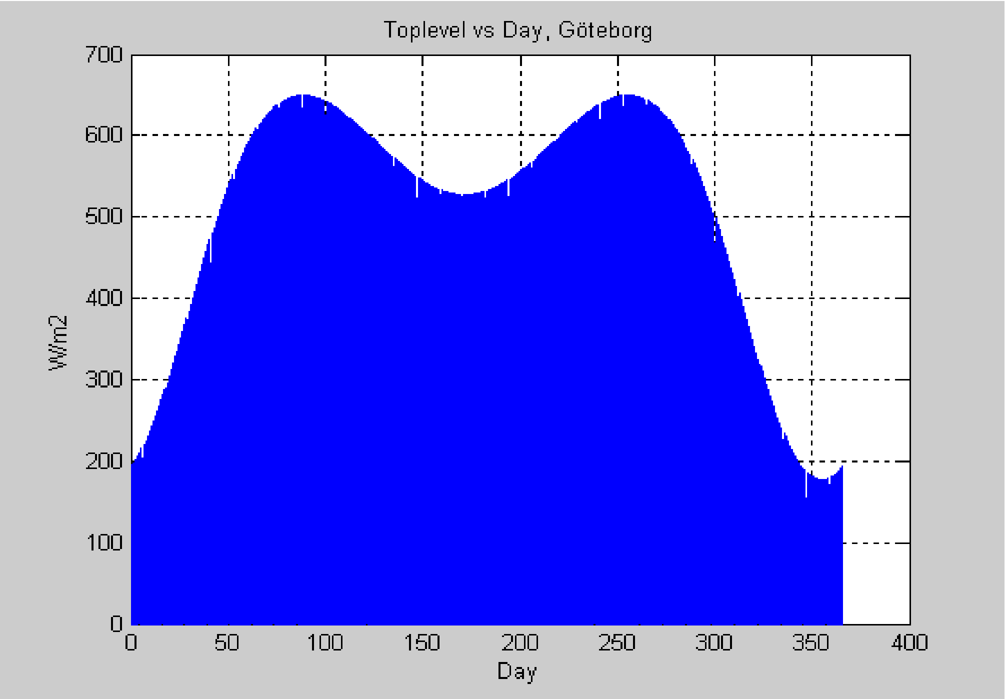

Figure 53 SRZ = 90 and SRS = 0. Toplevel vs day during a year. Göteborg

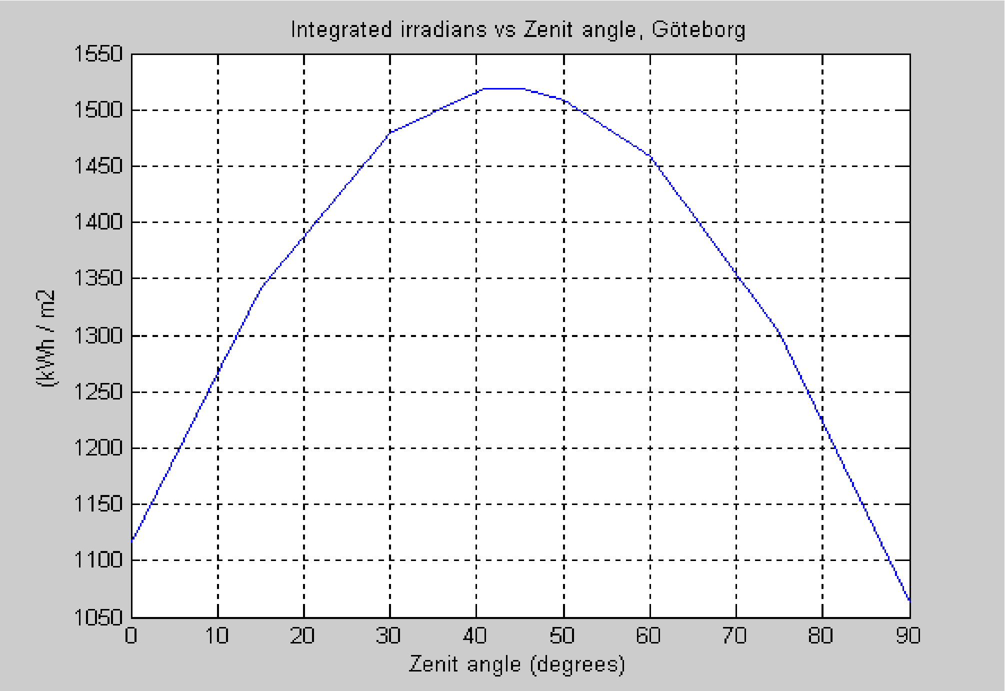

Figure 54 Integrated irradiance during a year vs Zenit angle (SRZ). SRS = 0. Göteborg

Surface_rel_Z	Integration (kWh / m^2)
0	837
15	1026
30	1152
35	1178
44	1204
45	1205
46	1206
47	1206
48	1206
49	1206
50	1205
51	1204
55	1197
60	1180
75	1078
90	908
“Sun Tracking”	1835

Table 27 Integrated irradiance during a year for different zenith angles and “Sun Tracking”. SRS = 0. Kiruna

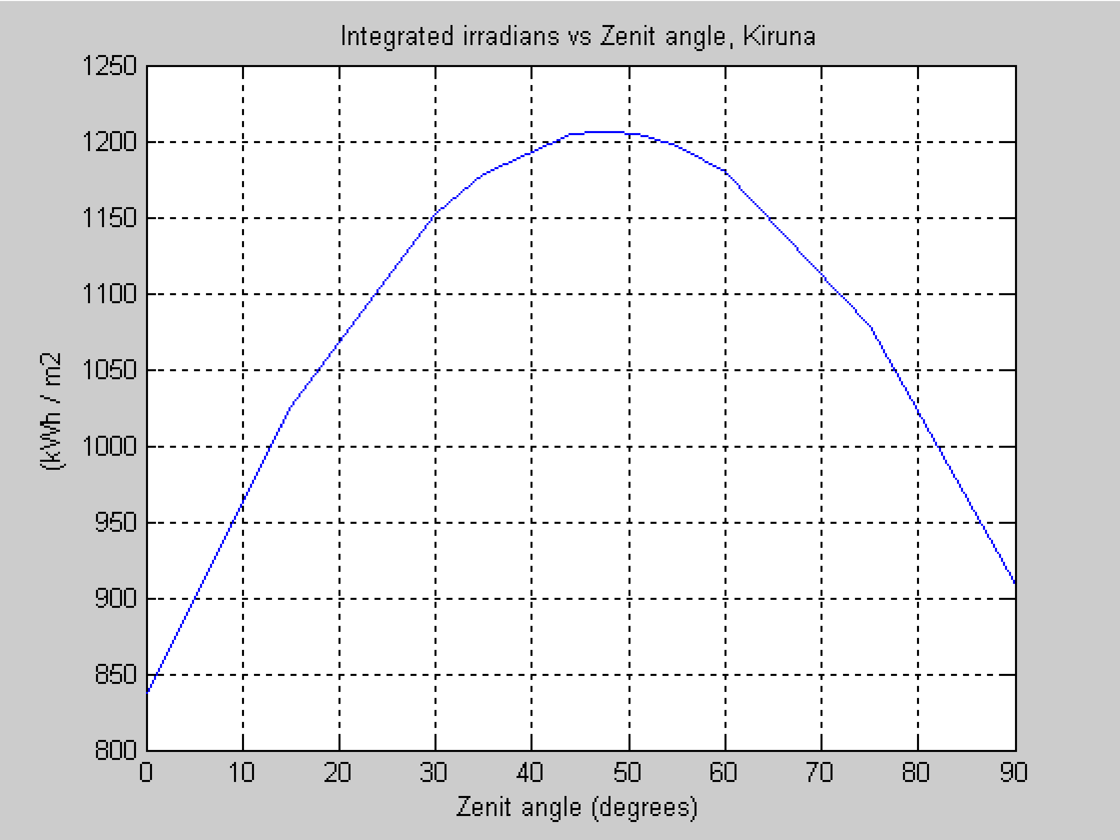

Figure 55 Integrated irradiance during a year vs Zenit angle (SRZ). SRS = 0. Kiruna

Surface_rel_Z	Integration (kWh / m^2) *Surface_rel_S* = 0	Integration (kWh / m^2) *Surface_rel_S* = 180
0	2323 (See Figure 57)	2323 (See Figure 57)
1	2322	2324
2	2320	2324
3	2317	2323
4	2313	2322
5	2309	2320
10	2277	2299
15	2228	2260
30	1981	2044
45	1604 (See Figure 58)	1692 (See Figure 59)
60	1136	1239
75	704	788
90	370 (See Figure 60)	433 (See Figure 61)
“Sun Tracking”	3187	3187

Table 28 Integrated irradiance during a year for different zenith angles (SRZ) and “Sun Tracking”. SRS = 0 and 180. Nairobi

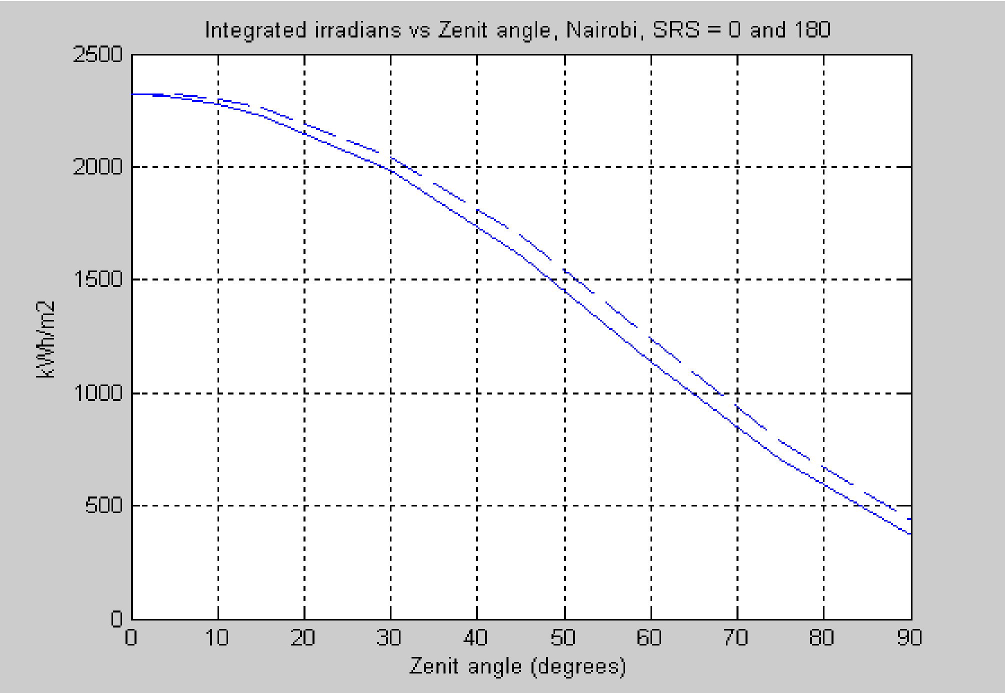

Figure 56 Integrated irradiance during a year vs Zenit angle (*Surface_rel_Z)*. Nairobi. SRS (Surface_rel_S) = 0 (____) respectively 180 (_ _ _)

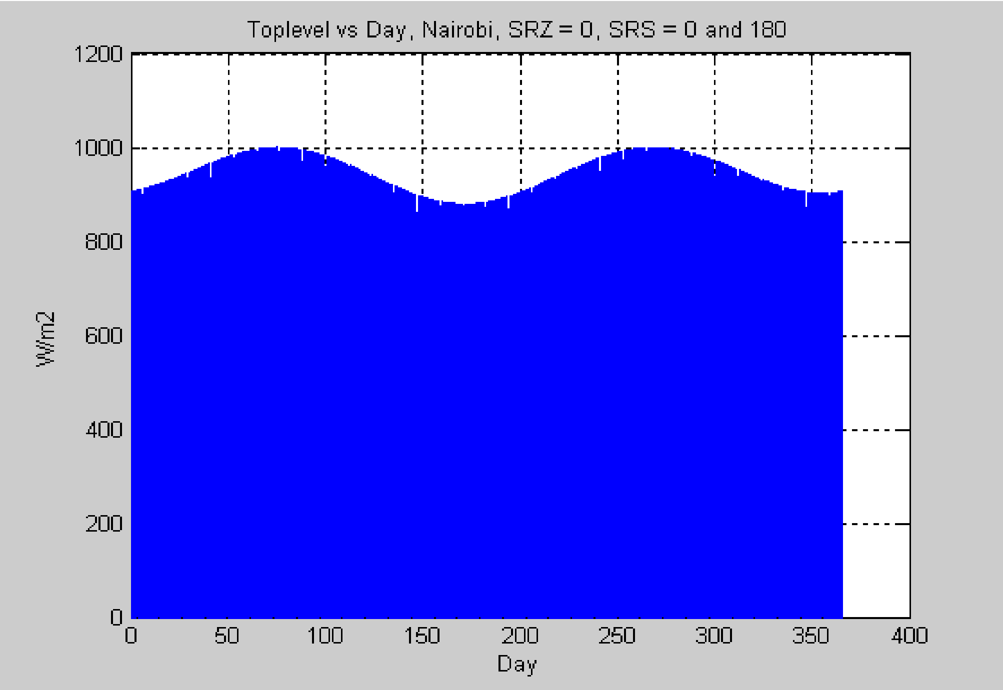

Figure 57 SRZ = 0 and SRS = 0 resp 180. Toplevel vs day during a year. Nairobi

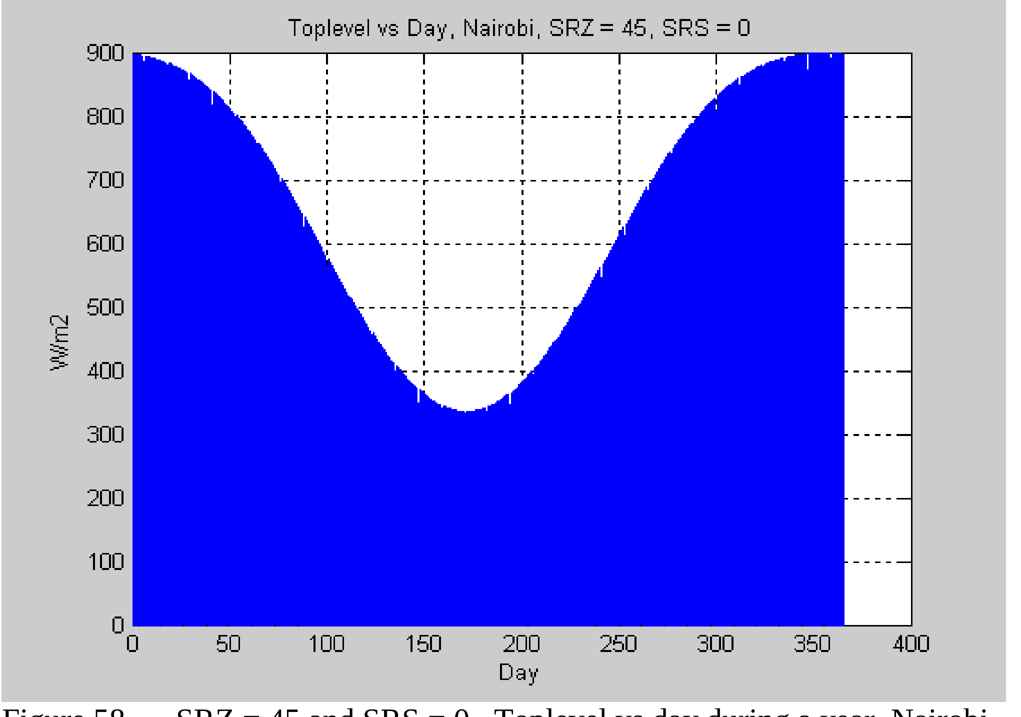

Figure 58	SRZ = 45 and SRS = 0. Toplevel vs day during a year. Nairobi

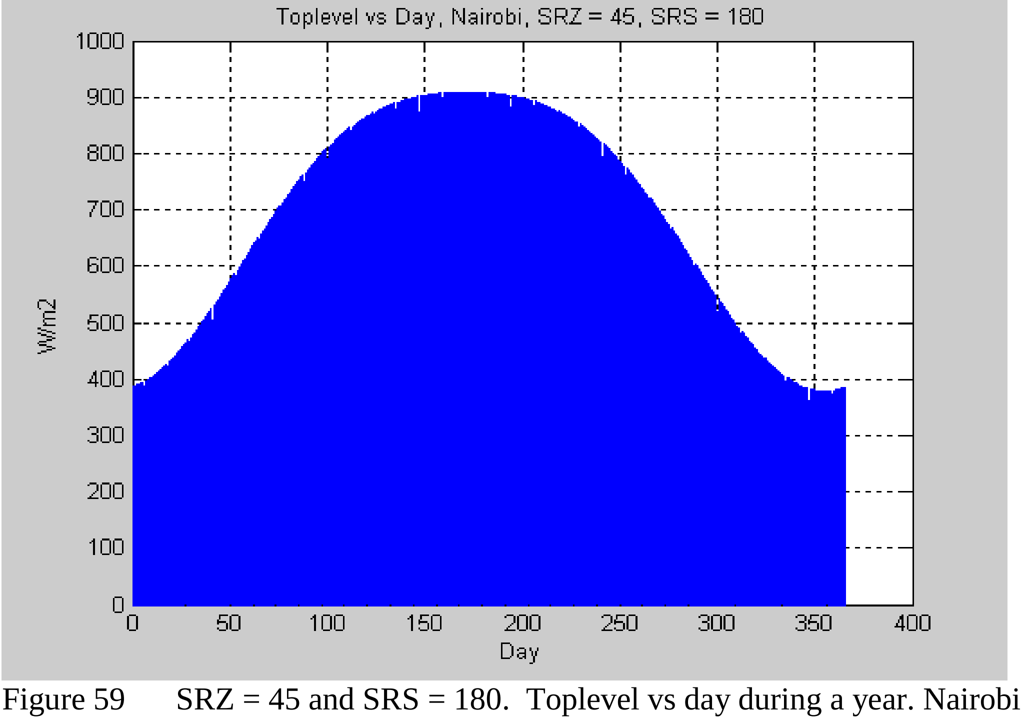

Figure 59	SRZ = 45 and SRS = 180. Toplevel vs day during a year. Nairobi

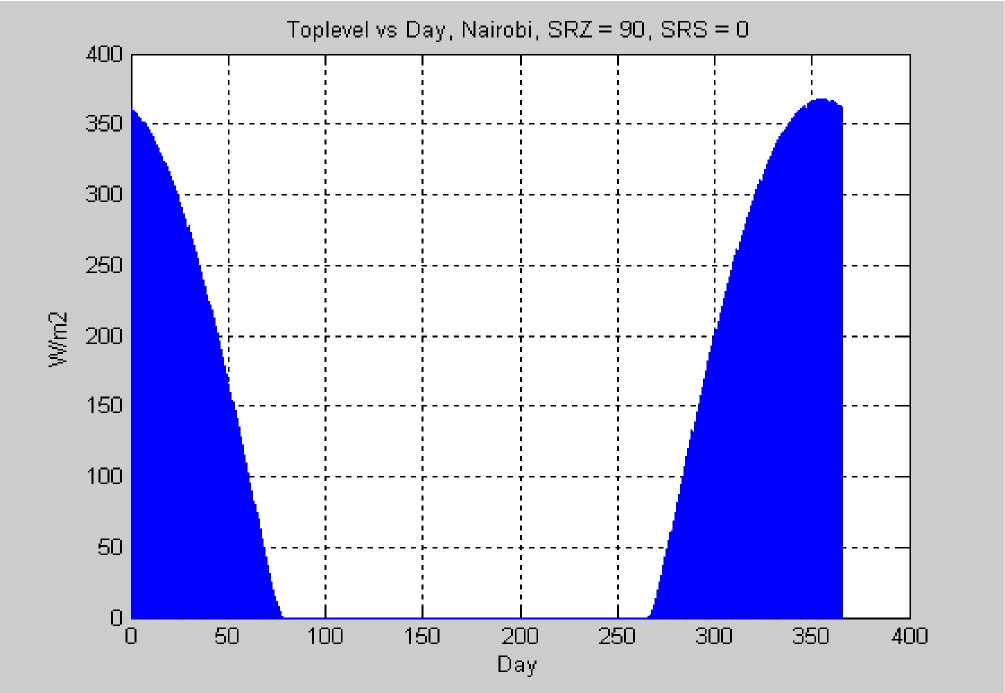

Figure 60 SRZ = 90 and SRS = 0. Toplevel vs day during a year. Nairobi

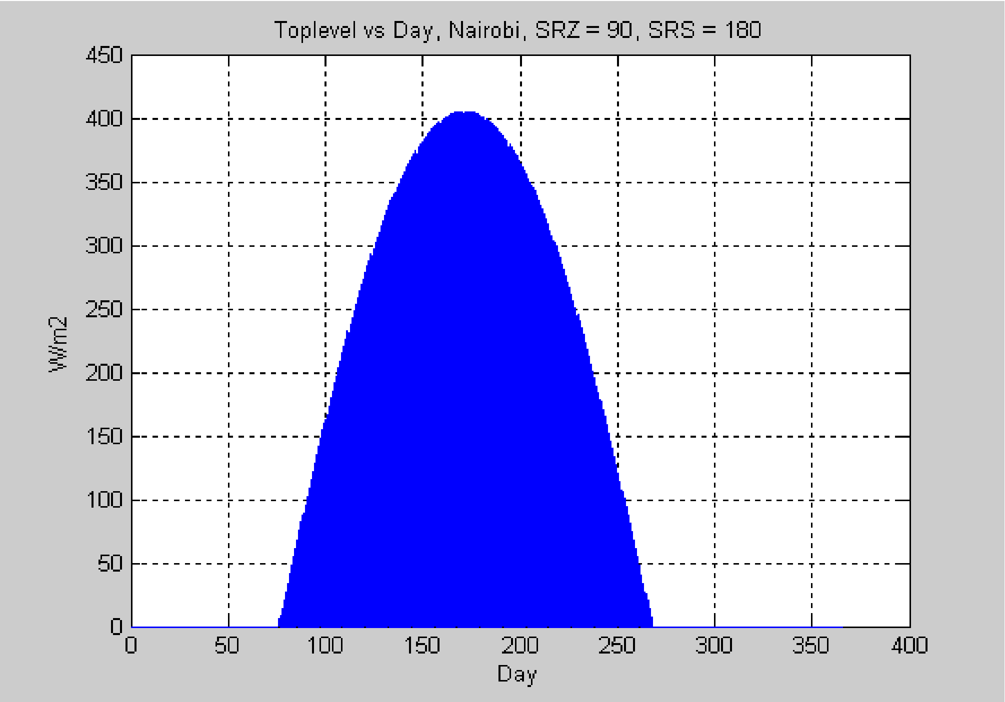

Figure 61 SRZ = 90 and SRS = 180. Toplevel vs day during a year. Nairobi

4.5 Sun_panel_generator

4.5.1 Principle

"Sun_panel_generator" is a program function (subroutine) that generates the electric power from a number of solar panels that are radiated by sun irradiation. The program function has as input an irradiation vector, named "*Radiation_vector*" that is generated by the routine "Sun_Intensity". The output from the routine "Sun_panel_generator" is a vector, called "*P_sun_el_vector*", with electric power values for the different time steps. The calculation is according the following:

Equation 45:

$$P\_sun\_el\_vector(time\_step) = Radiation\_vector(time\_step) \cdot Sun\_panel\_area \cdot$$

$$\cdot Power\_factor\_cells \cdot Power\_factor\_MPP \cdot Power\_factor\_electr / 1000 \quad (kW)$$

Where:

P_sun_el_vector: vector with generated power from the solar cells generator (kW)

time_step: current time step (dimension less)

Radiation_vector: vector with irradiation values (W/m^2)

Sun_panel_area: total solar cell area (m^2)

Power_factor_cells: The efficiency of the solar cells (radiated power to electric power) (0 - 1)

Power_factor_MPP: The efficiency to adapt the working point to the most effective one, called Maximum Power Point (0 – 1). A perfect adaption, results in *Power_factor_MPP* = 1

Power_factor_electr: The efficiency of the power electronics (0 – 1)

4.5.2 Input parameters

The input parameters of the routine are specified in Table 29.

Parameter name	Purpose
Radiation_vector Input via function argument from Main Program	Vector with irradiation (effective to the solar cells) (per simulation step)
Power_factor_cells	The efficiency of the solar cells (radiated power to electric power) (0 - 1)
Power_factor_MPP	The efficiency to adapt the working point to the most effective one, called Maximum Power Point (0 – 1). A perfect adaption, results in *Power_factor_MPP* = 1
Power_factor_electr	The efficiency of the power electronics (0 – 1)
Sun_panel_area	total solar cell area $\left(m^2\right)$

Table 29 Input parameters for routine “Sun_panel_generator”

4.5.3 Output parameters

The output parameters of the routine are specified in Table 30.

Parameter name	Purpose
P_sun_el_vector Output via function argument from Main Program	Vector with electric sun power (per simulation step) (kW)

Table 30 Output parameters for routine “Sun_panel_generator”

4.5.4 Examples

Table 31 gives an example of used parameters in a simulation.

Parameter name	**Used value in the example**
Power_factor_cells	0.15
Power_factor_MPP	0.95
Power_factor_electr	0.95
Sun_panel_area	800

Table 31 Table 31 gives an example of used parameters in a simulation.

The resulted output power from the Sun power generator with values according to Table 31 and irradiances as in Figure 62 follow in Figure 63.

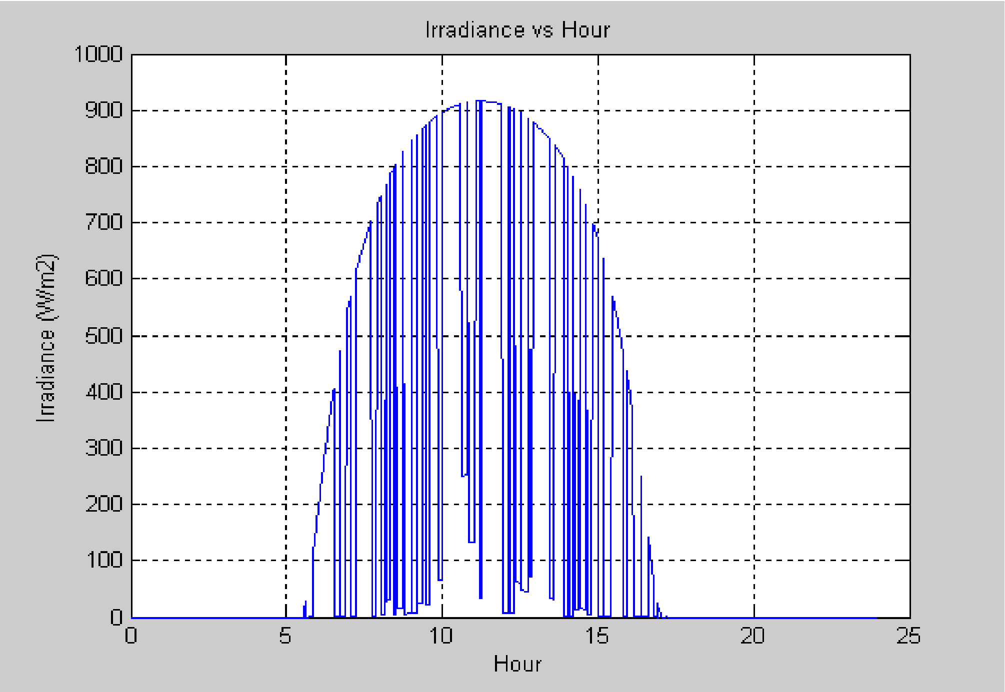

Figure 62 An example of irradiances during a period of 24 hours

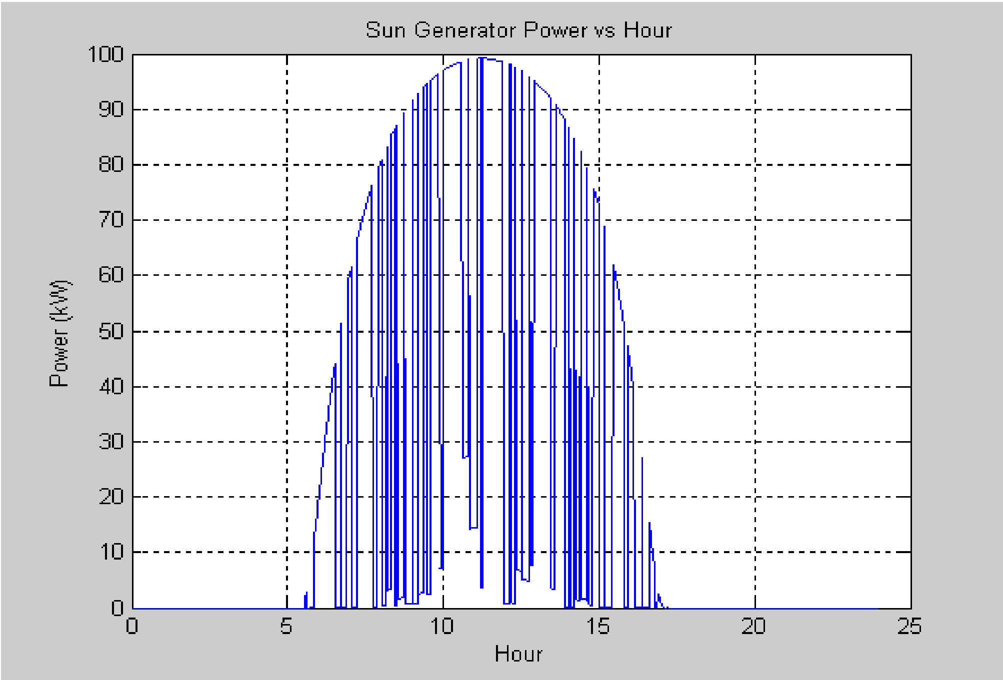

Figure 63 The resulted power from the sun power generator when using the parameters in Table 31 together with the irradiances according to Figure 62

4.6 Load_make

4.6.1 Principle

4.6.1.1 Common

“Load_make” is a program function (subroutine) that generates a stochastic model of a grid load that is connected to the Wind generator(s), the Sun generator(s) and the energy storage. See [3]. This load corresponds to the load of the consumer grid.

4.6.1.2 Base values and day time points

The routine works with 6 separated day time points, TP_0 – TP_5. TP_0 is defined as midnight (0 or 24). TP_1 – TP_5 should be defined as input values. The routine also uses 4 primary base power levels, A_prim, B_prim, C_prim and D_prim, that are related to the time points according to:

- A_prim: base power at TP_0
- B_prim: base power at TP_1 to TP_2
- C_prim: base power at TP_3 to TP_4
- D_prim: base power at TP_5

That means that the primary base levels follow the following sequence for a 24 hours cycle:

A_prim(at TP_0) – B_prim (at TP_1) – B_prim (at TP_2) – C_prim(at TP_3) –

- C_prim(at TP_4) – D_prim(at TP_5) – A_prim(back to a new 24 hours cycle, TP_0).

The shifts between the 4 base power levels are performed lineary.

4.6.1.3 Stochastic variations

4.6.1.3.1 Low frequency variations

The 4 primary base power levels undergo a stochastic variation according a normal process. This stochastic variation are performed once per 24 hours cycle. This variation is named "the low frequency" variation. The statistic parameters are named *"L_My_L"* and *"L_sigma_L"* (the first L stands for Load and the second stands for Low), representing the mean value respectively the standard deviation in the Normal distribution. The so called "*Load_factor*" that is achieved by the Normal distribution and that is generated once per 24 hours cycle is used as a factor for all base levels. This means that the base levels are updated every new 24 hours cycle according to Equation 46:

Equation 46. (1 – 4):

1. $A = A\_prim \cdot Load\_factor$

2. $B = B\_prim \cdot Load\_factor$

3. $C = C\_prim \cdot Load\_factor$

4. $D = D\_prim \cdot Load\_factor$

The values of A, B, C and D are the real (used) base values during the present 24 hours cycle. The next coming 24 hours cycle will result in a new set of base levels (from the original levels of A_prim, B_prim, C_prim and D_prim) and so on.

Figure 64 illustrates the principle of building up the base power structure during a 24 hours cycle.

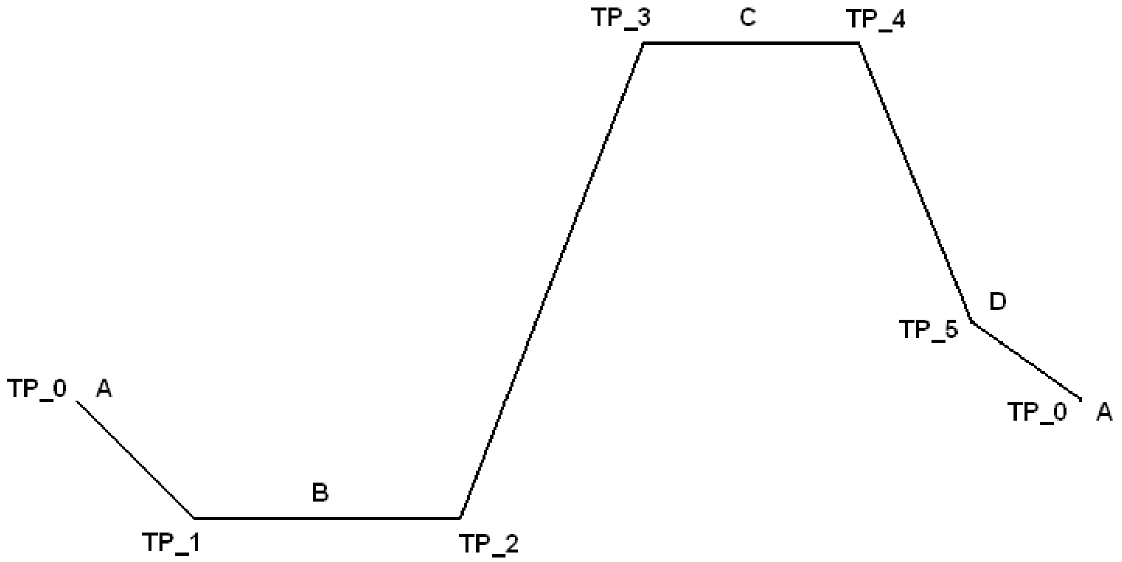

Figure 64 An illustration of the base power shifts between the different time points during a 24 hours cycle

4.6.1.3.2 High frequency variations

For each simulation step a “Normal distributed noise” is added to the present level that is achieved by the base levels. This “high frequency” variation has the statistic parameters *“L_My_H”* and *“L_sigma_H”* (the L stands for Load and the H stands for High), representing the mean value respectively the standard deviation in the Normal distribution.

An example of this “Power Noise” effect is illustrated in Figure 65 (no “Power Noise added) and Figure 66 (“Power Noise” added).

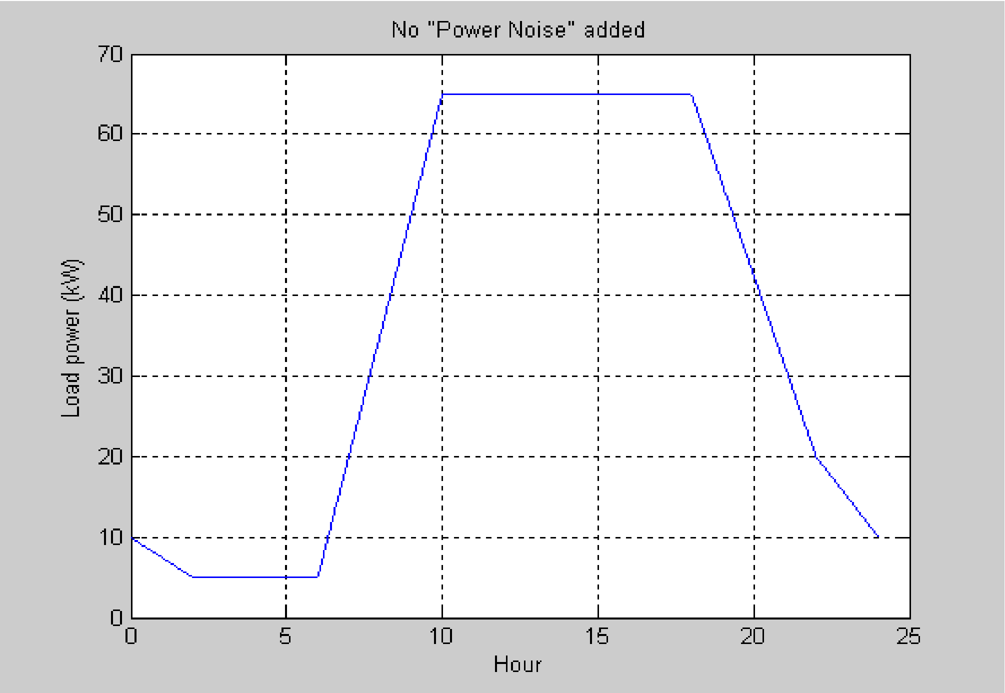

Figure 65 An example with **no** power noise added to the base levels

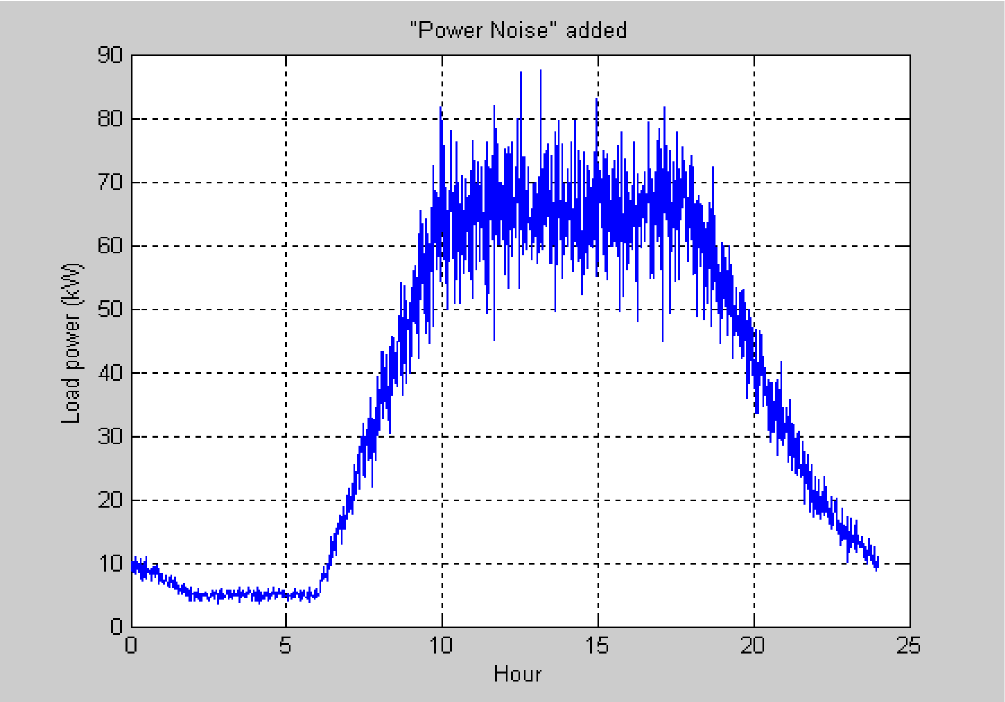

Figure 66 An example **with** power noise added to the base levels. The standard deviation *"L_sigma_H"* (in the example) is 10 % of present level without noise. The noise mean value *"L_My_H"* is zero.

4.6.1.4 Load Category a respectively b

The load model works with 2 categories of loads:

- Category a
- Category b

The simulation starts with category a. This category will be modeled corresponding to a specified number of days, "*a_limit*". After this is completed, follows simulation with category b. This category is modeled corresponding to a specified number of days, "b_limit". This sequence is then repeated as long as the total simulation continues. The dividing into 2 separated categories is useful when modelling for example the power consumption in e.g. an industry area. In this case it could be a large difference between the power profile during the working week compared with the weekend.

The effect of this separating into 2 categories is illustrated in Figure 67. The figure show the load during 10 days. The first 5 days correspond to a working week. Then follow 2 days corresponding to the weekend, and so on.

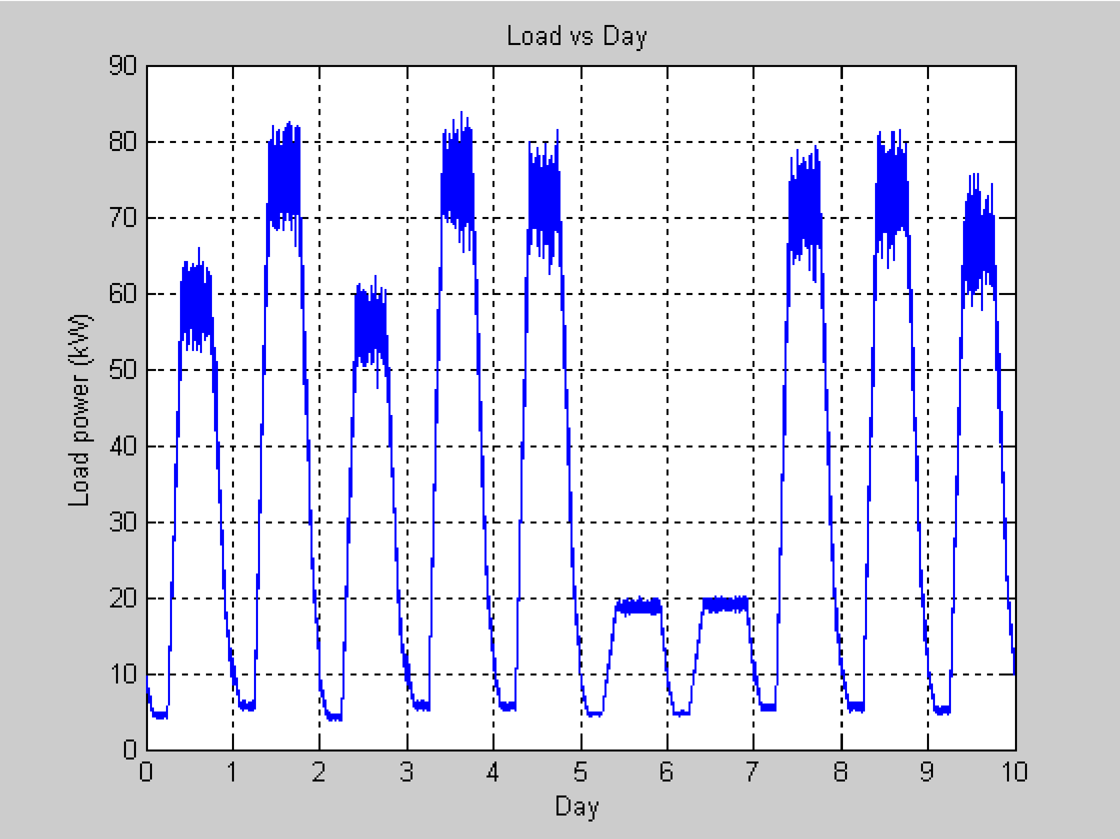

Figure 67 A simulation sequence corresponding to 5 working days followed by 2 days of weekend

4.6.2 Input parameters

The input parameters of the routine are specified in Table 32.

Parameter name	Purpose
Sim_step_total Input via function argument from Main Program	The total number of simulation steps per sequence (Sim_step_total = 43200 corresponds to a simulation sequence over a time of 30 days if Sim_step_sec = 60)
Sim_step_sec Input via function argument from Main Program	Time interval in seconds per simulation step (60 is a standard value)
a_limit	Number of days for modeling according category a, before change to category b. The simulation always starts with category a
b_limit	Number of days for modeling according category b, before change to category a. The simulation always starts with category a
A_prim_a	This relates to load category a. For further definition see 4.6.1.2.
B_prim_a	-“-
C_prim_a	-“-
D_prim_a	-“-
A_prim_b	This relates to load category b. For further definition see 4.6.1.2.
B_prim_b	-“-
C_prim_b	-“-
D_prim_c	-“-
TP_1_a	This relates to load category a. For further definition see 4.6.1.2.
TP_2_a	-“-
TP_3_a	-“-
TP_4_a	-“-
TP_5_a	-“-
TP_1_b	This relates to load category b. For further definition see 4.6.1.2.
TP_2_b	-“-
TP_3_b	-“-
TP_4_b	-“-
TP_5_b	-“-

L_My_H_a	This relates to load category a. For further definition see 4.6.1.3.2.
L_Sigma_H_a	-“-
L_My_L_a	This relates to load category a. For further definition see 4.6.1.3.1..
L_Sigma_L_a	-“-
L_My_H_b	This relates to load category b. For further definition see 4.6.1.3.2.
L_Sigma_H_b	-“-
L_My_L_b	This relates to load category b. For further definition see 4.6.1.3.1.
L_Sigma_L_b	-“-
Load_power_file	The name of a *Load power file* (string) to store the load vector and the above parameters

Table 32 Input parameters for routine “Load_make”

4.6.3 Output parameters

The output parameters of the routine are specified in Table 33.

Parameter name	**Purpose**
P_load_vector Output via function argument to Main Program	Vector with load power (per simulation step)
Load_power_file	See Table 32

Table 33 Output parameters for routine “Load_make”

4.6.4 Examples

Table 34 gives examples of used parameters in a simulation.

Parameter name	Used values in the example
Sim_step_total Input via function argument from Main Program	43200
Sim_step_sec Input via function argument from Main Program	60
a_limit	5
b_limit	2
A_prim_a	10 (kW)
B_prim_a	5 -“-
C_prim_a	65 -“-
D_prim_a	20 -“-
A_prim_b	10 -“-
B_prim_b	5 -“-
C_prim_b	20 -“-
D_prim_b	20 -“-
TP_1_a	2 (hour)
TP_2_a	6 -“-
TP_3_a	10 -“-
TP_4_a	18 -“-
TP_5_a	22 -“-
TP_1_b	2 (hour)
TP_2_b	6 -“-
TP_3_b	10 -“-
TP_4_b	18 -“-
TP_5_b	22 -“-
L_My_H_a	0.0
L_Sigma_H_a	0.04
L_My_L_a	1.0
L_Sigma_L_a	0.15
L_My_H_b	0.0
L_Sigma_H_b	0.02
L_My_L_b	1.0
L_Sigma_L_b	0.10
Load_power_file	‘Load_1’

Table 34 Examples of used input parameters for routine “Load_make”

The simulation result regarding the generated load power based on parameters according to Table 34 follows in Figure 68.

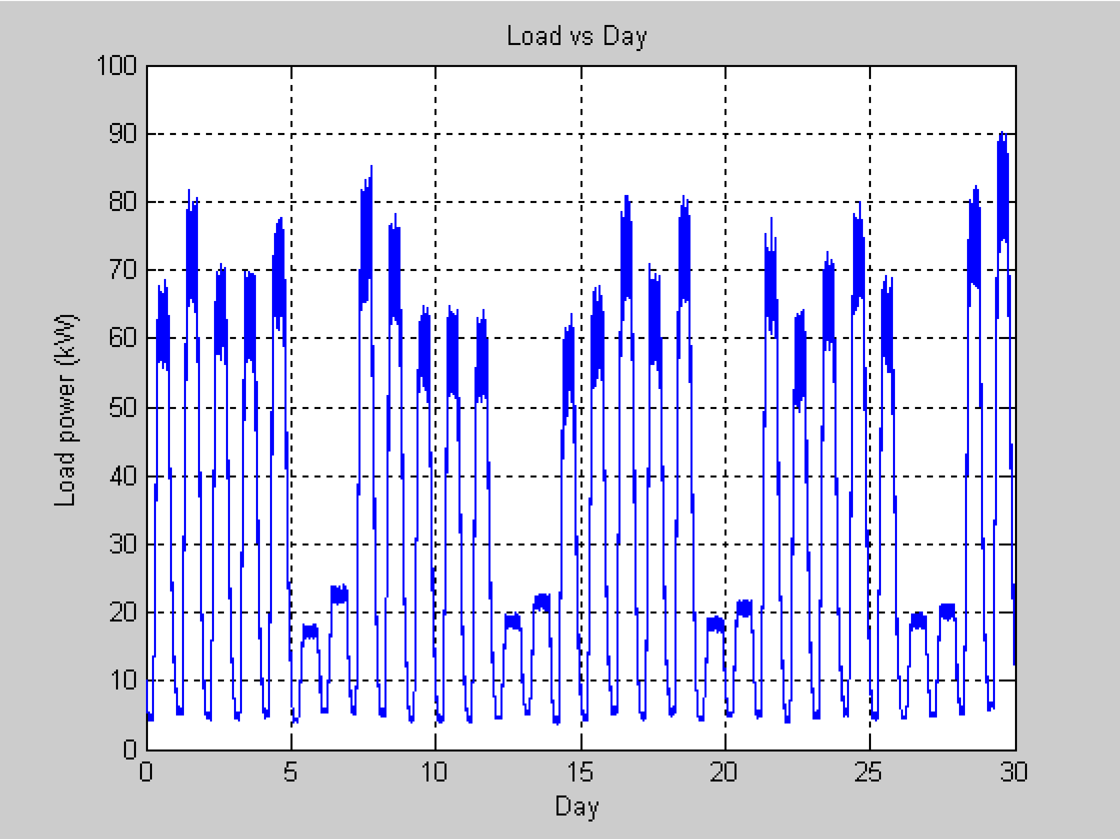

Figure 68 The loaded power for a simulation sequence of 30 days. Input parameters according to Table 34

4.7 Connect_Gen_load

4.7.1 Principle

“Connect_Gen_load” is a program function (subroutine) that models the electrical connection of the Windpower and Sunpower generators to a loaded (consumer) grid. Two separated types of power levels have been defined:

- Gross power
- Net power

4.7.1.1 Gross power

The *gross power* is the sum of the electrical power generated by the Wind power generator(s) and the Sun power gerator(s).

The sum of the electrical power generated by the Wind power generator(s) and the Sun power gerator(s) (the gross power) is for each single simulation step loaded in a vector named “*P_generation_vector*”.

4.7.1.2 Net power

The *net power* is the *gross power* minus the *loaded power* from the loaded grid.

The *net power* is for each single simulation step loaded in a vector named *“P_buffer_vector”*.

4.7.2 Input parameters

The input parameters of the routine are specified in Table 35.

Parameter name	**Purpose**
P_wind_el_vector (Input via function argument from Main Program)	Vector with electric power generated by the wind generator(s) (for each single simulation step)
P_sun_el_vector (Input via function argument from Main Program)	Vector with electric power generated by the sun generator(s) (for each single simulation step)
P_load_vector (Input via function argument from Main Program)	Vector with electric loaded power (from the consumer grid) (for each single simulation step)

Table 35 Input parameters for routine “Connect_Gen_load”

4.7.3 Output parameters

The output parameters of the routine are specified in Table 36.

Parameter name	Purpose
P_generation_vector (Output via function argument to Main Program)	See paragraph 4.7.1.1.
P_buffer_vector (Output via function argument to Main Program)	See paragraph 4.7.1.2.

Table 36 Output parameters for routine “Connect_Gen_load”

4.7.4 Examples

4.7.4.1 Example 1 (24 hours simulation)

The following example is based on a single simulation during a time sequence of 24 hours. Assumptions according to the following was done:

- Local area: Göteborg
- Time: 20 June
- Cloudiness: 1/8
- Weibull parameters: A = 7, C = 2
- Number of wind generators corresponding to “Hönö”: 3
- Solar panel area: 500 m^2
- Sun Tracking: yes

One single **sample** of simulation resulted in the following:

- Mean wind rate: 7.6 m/sec
- Standard deviation wind rate: 1.3 m/sec

- Mean extinction coefficient: 0.41
- Standard deviation extinction coefficient: 0.13

The loaded power was simulated as shown in Figure 69.

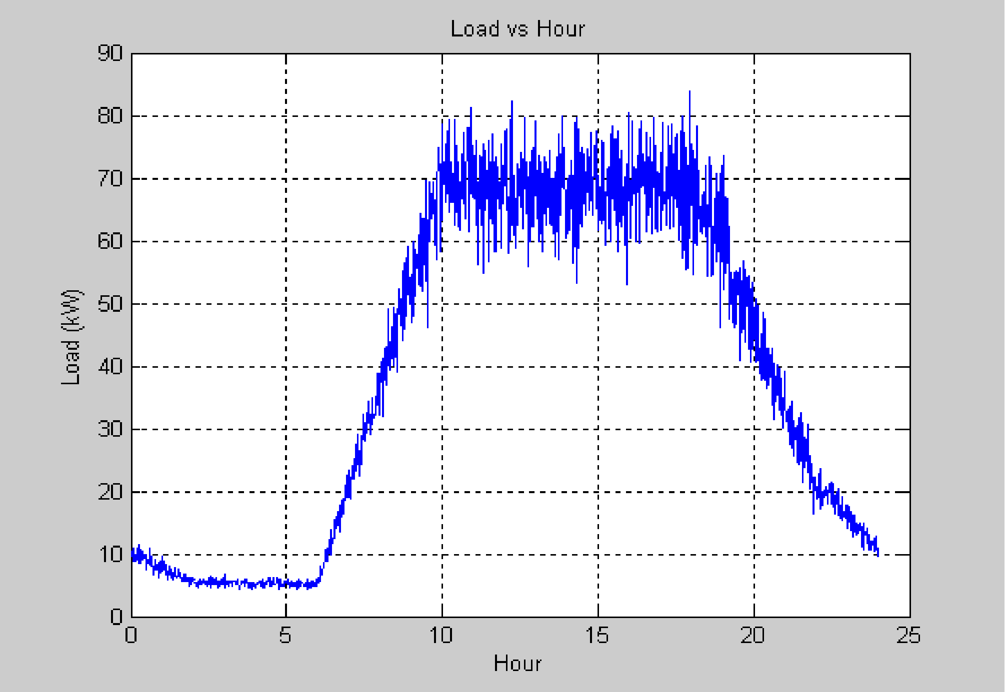

Figure 69 Loaded power. Example 1

Based on above defined conditions, gross- respectively net power according to Figure 70 respectively Figure 71 were obtained. Figure 72 and Figure 73 show the separated wind- and sun power that were generated.

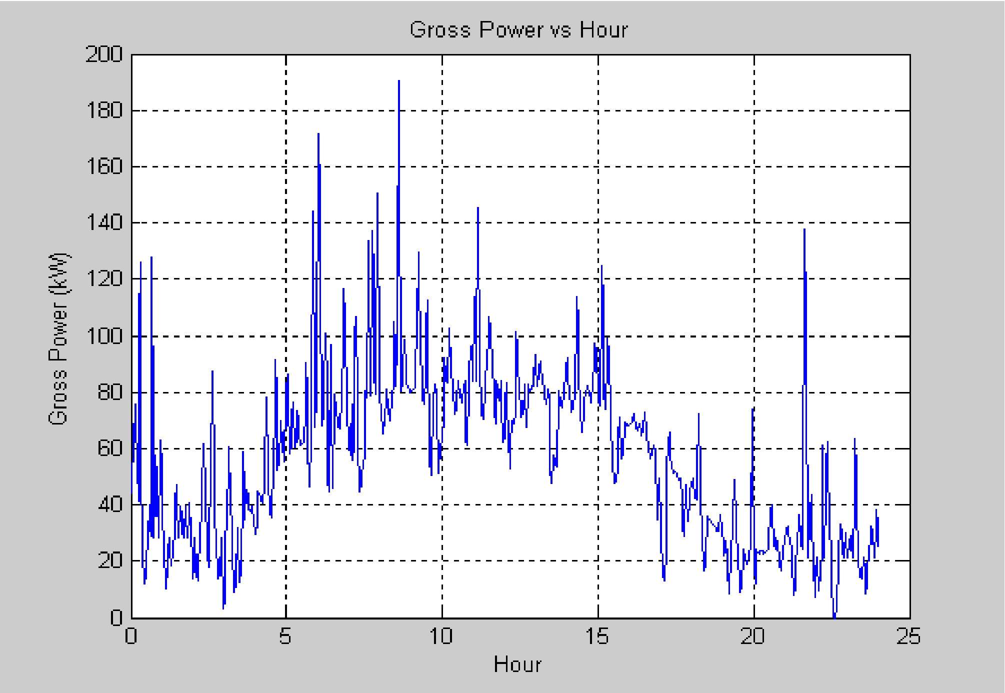

Figure 70 Generated gross power. Example 1

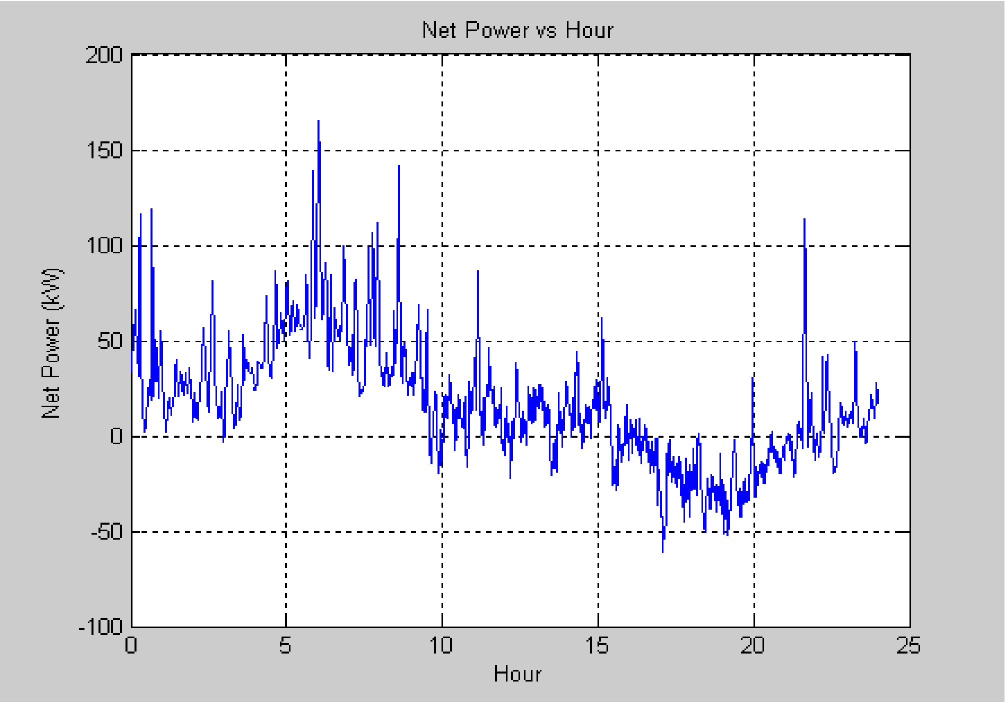

Figure 71 Generated net power. Example 1

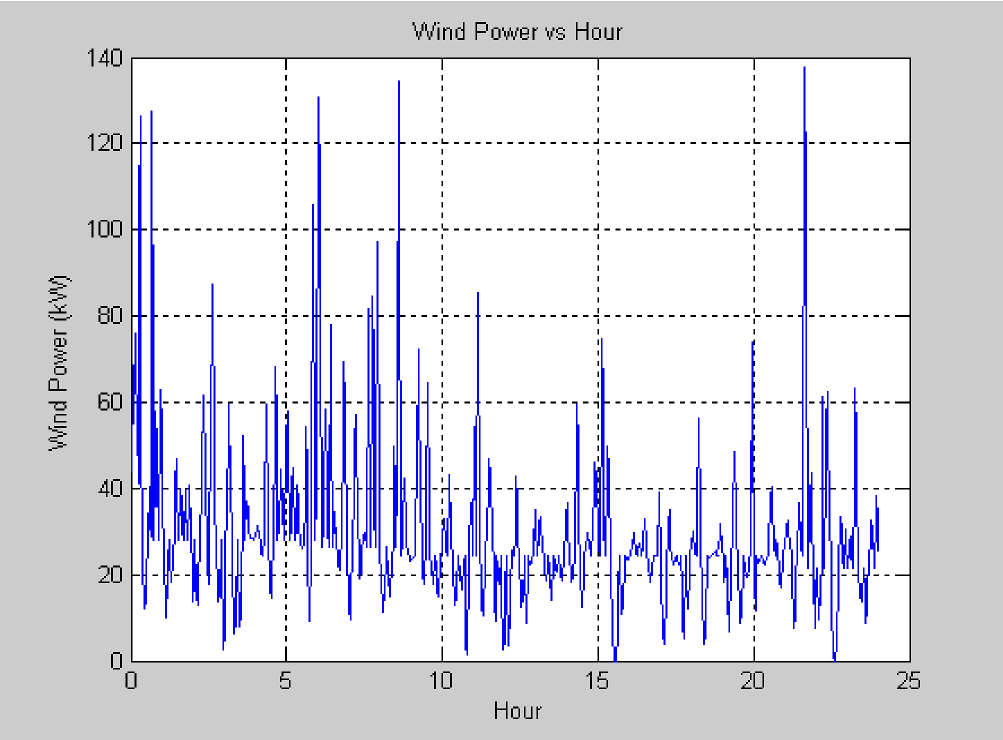

Figure 72 Generated Wind power. Example 1

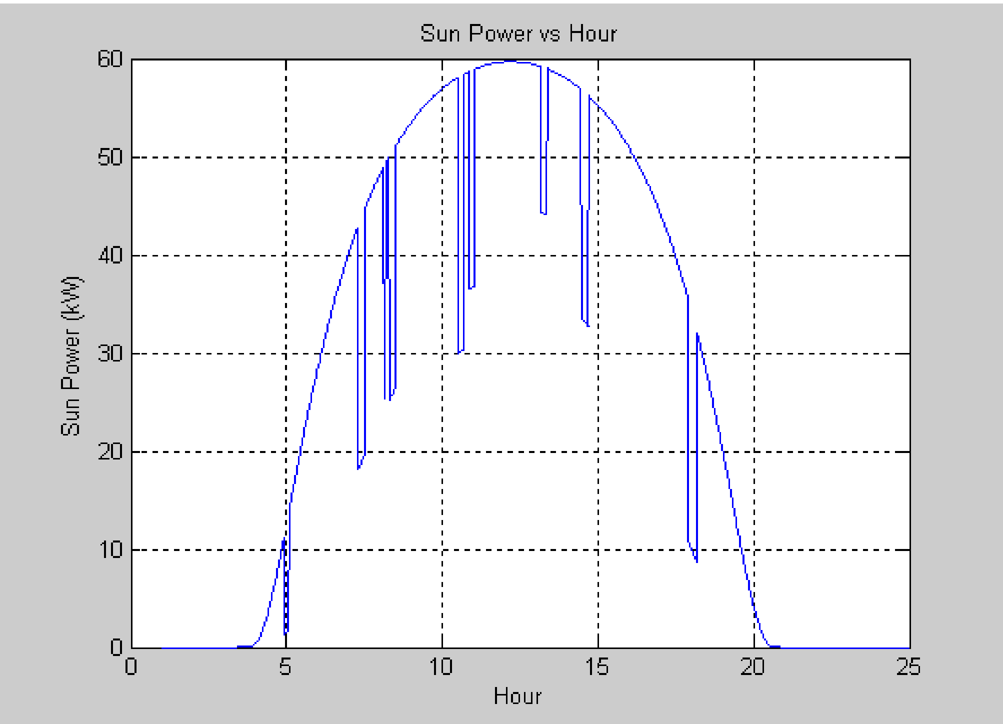

Figure 73 Generated Sun power. Example 1

4.7.4.2 Example 2 (10 days simulation)

This example is based on a single simulation during a time sequence of 10 days (240 hours). The other assumptions agree with example 1.

One single **sample** of simulation resulted in the following:

- Mean wind rate: 3.5 m/sec
- Standard deviation wind rate: 1.8 m/sec
- Mean extinction coefficient: 0.40
- Standard deviation extinction coefficient: 0.13

The loaded power was simulated as shown in Figure 74.

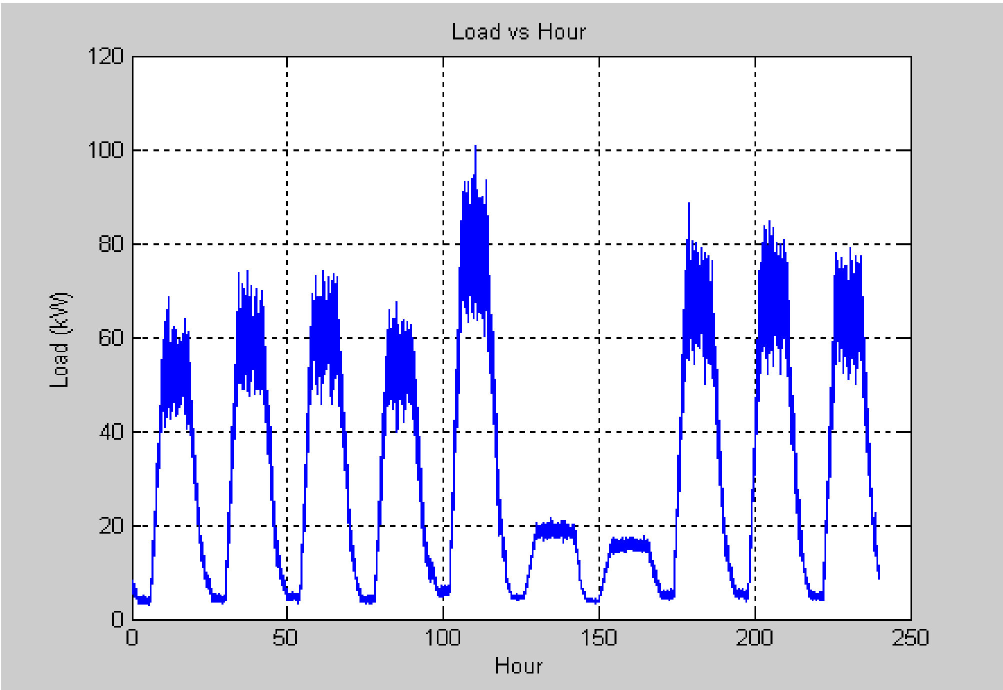

Figure 74 Loaded power. Example 2

Based on above defined conditions, gross- respectively net power according to Figure 75 respectively Figure 76 were obtained. Figure 77 and Figure 78 show the separated wind- and sun power that were generated.

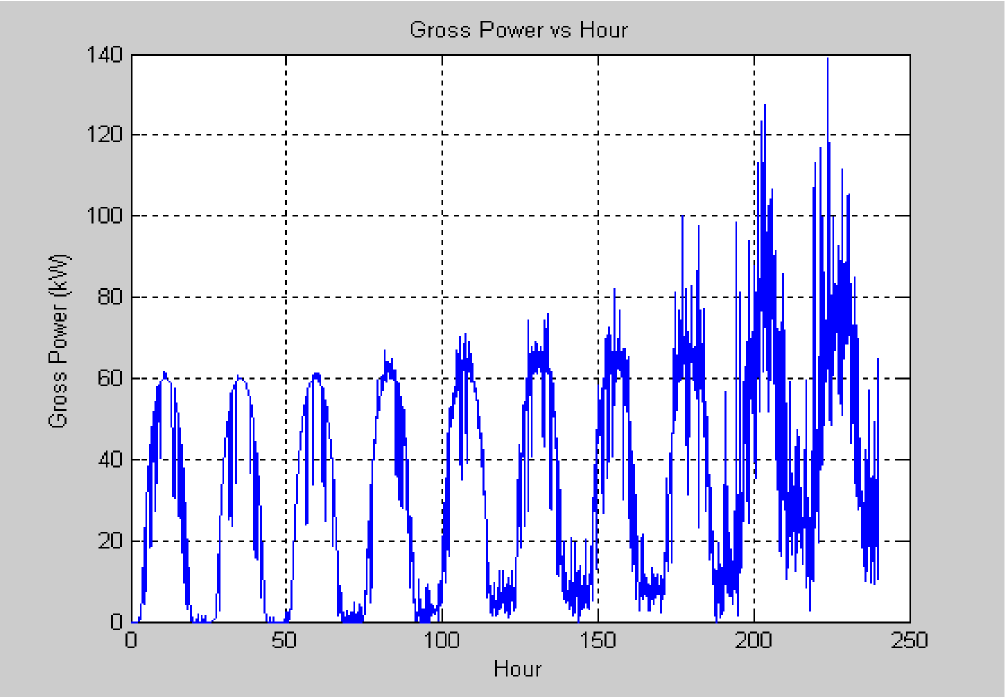

Figure 75 Generated gross power. Example 2

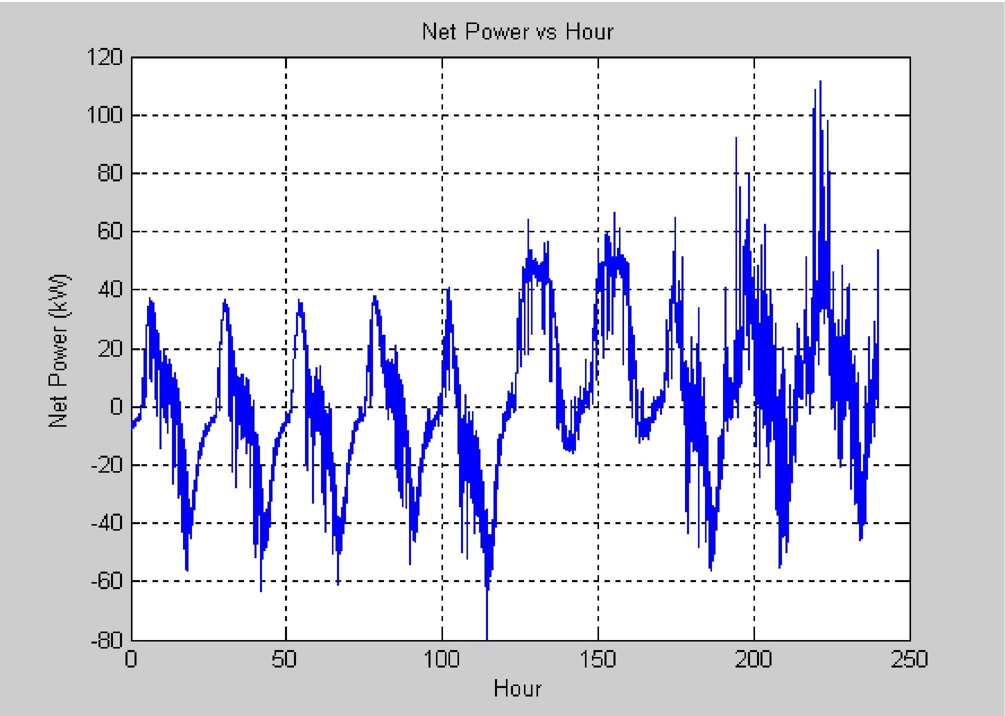

Figure 76 Generated net power. Example 2

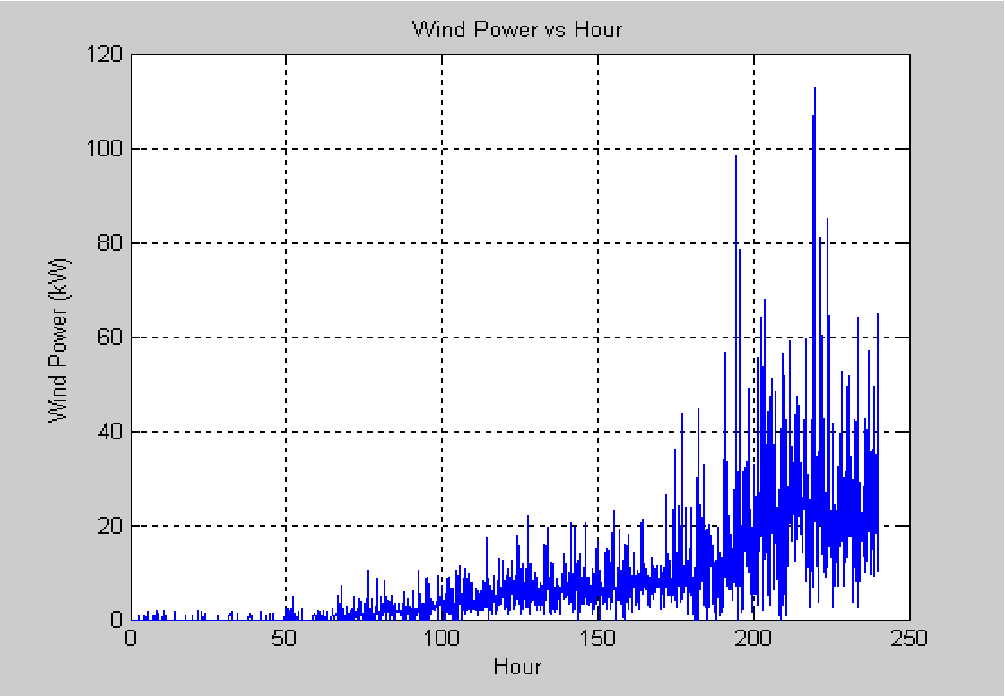

Figure 77 Generated Wind power. Example 2

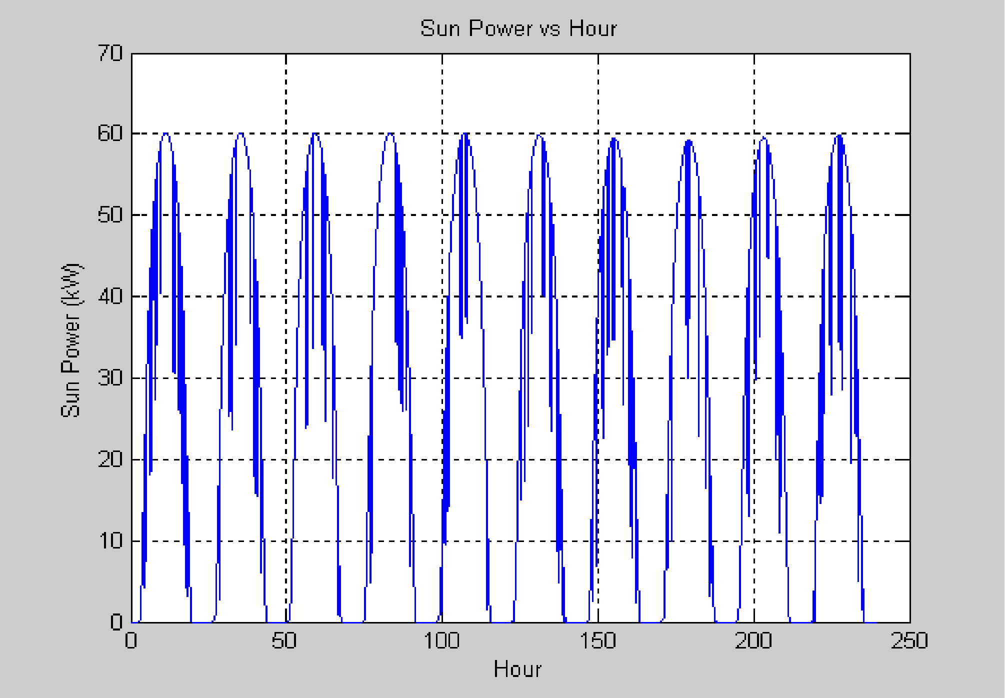

Figure 78 Generated Sun power. Example 2

4.8 Battery_Distribution

4.8.1 Principle

"Battery_Distribution" is a program function (subroutine) that models the electrical function of the energy storage (battery) and the distribution grid. The function (subroutine) get inputs from the function (subroutine) "Connect_Gen_load". Two separated modes have been defined:

- Storing / loading of energy to / from the energy storage
- Exporting / importing of energy to / from the distribution grid

4.8.1.1 Storing / Loading of energy to / from the energy storage

This mode is in question if one of the following 2 conditions are fulfilled:

Condition 1:

a) *Net power* > *0* (see paragraph 4.7.1.2)

And

b) *battery_charge(time_step)* < *charge_level*_max

Condition 2:

a) *Net power* < *0* (see paragraph 4.7.1.2)

And

b) *battery_charge(time_step)* > *charge_level*_min

where

battery_charge(time_step):	current battery charge
*charge_level*_max:	maximum allowed charge level in the energy storage
*charge_level*_min:	minimum allowed charge level in the energy storage
time_step:	current simulation step

If **“Condition 1”** is fulfilled then the energy corresponding to the *Net power* multiplied with the time step of one simulation step, *time_step*, shall be **stored** in the energy storage. This energy is the part of produced energy that for the current simulation step not is consumed by the ordinary (consumer) grid load (Load_make. See paragraph 4.6). There is in other word a surplus energy on hand according to:

$$Surplus\_energy = Net\_Power \times time\_step$$

If **“Condition 2”** is fulfilled then the energy corresponding to the absolute value of the *Net power* multiplied with the time step of one simulation step shall be **loaded** from the energy storage. This energy is the part of consumed energy (by the consumer grid) that for the current simulation step not is produced by the ordinary generators (Wind_turbine see paragraph 4.2 together with Sun_panel_generator see paragraph 4.5). There is in other word a deficiency of energy on hand according to:

$$Deficiency\_energy = \left| Net\_Power \right| \times time\_step$$

Some other parameters regarding the energy storage

- Efficiency of storing (*charge_factor*)
- Efficiency of loading (*load_factor*)
- Maximum power for storing / loading (*power_max_batt*)
- Charge level in the energy storage at simulation start (*charge_level_init*)
- Self-discharge (*discharge_self*)

Figure 79 and Figure 80 illustrate the energy flow for “*Condition 1*” and “*Condition 2*”.

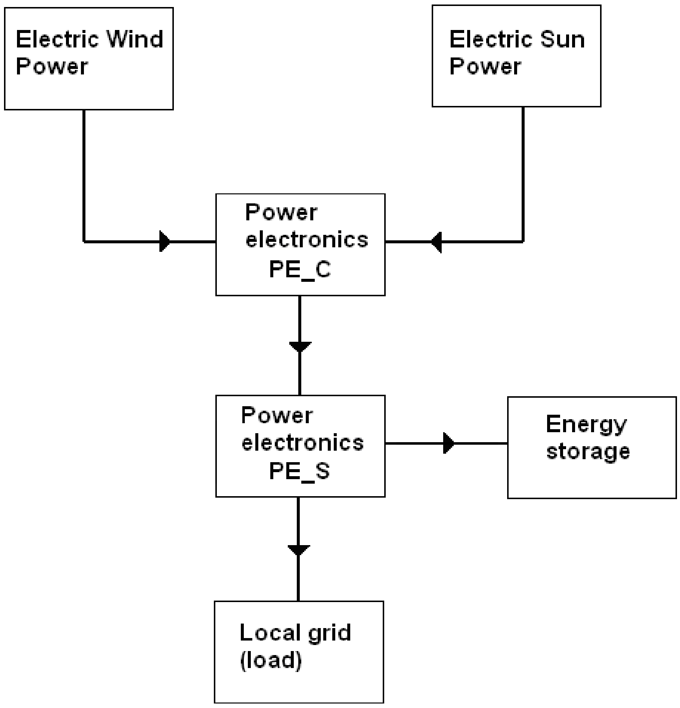

Figure 79 Energy flow for “*Condition 1*”. “Local grid” in the figure corresponds to “Consumer grid”

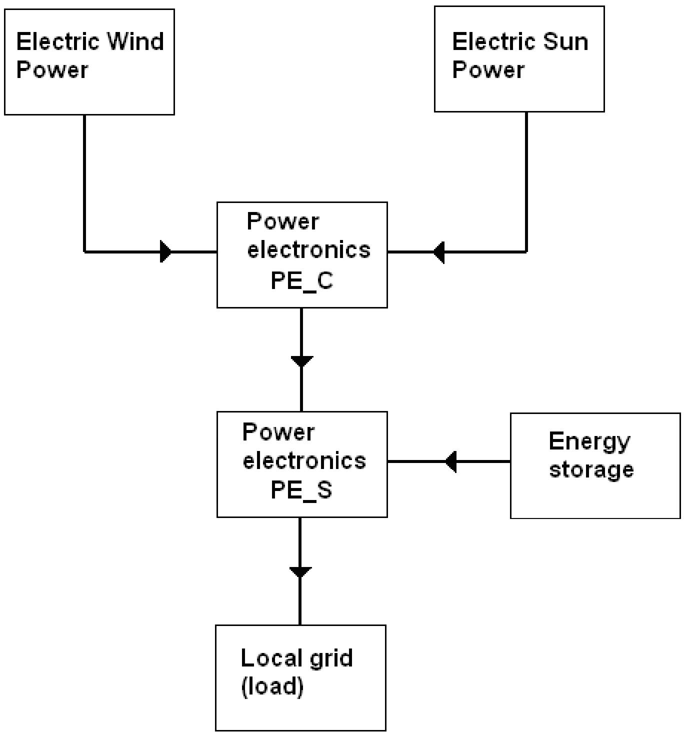

Figure 80 Energy flow for “*Condition 2*”. “Local grid” in the figure corresponds to “Consumer grid”

4.8.1.2 Exporting / importing of energy to / from the distribution grid

This mode is in question if **non** of the conditions in paragraph 4.8.1.1, Condition 1 and Condition 2, is fulfilled. Two conditions are then in question, named Condition 3 and condition 4:

<u>Condition 3</u>:

Condition 1.a is fulfilled

Condition 1.b is **not** fulfilled

Then **export** of generated surplus energy to the distribution grid shall be realized.

<u>Condition 4</u>:

Condition 2.a is fulfilled

Condition 2.b is **not** fulfilled

Then **import** of deficit energy from the distribution grid shall be realized.

The distribution grid is limited regarding maximum power for export / import by a parameter named *power_max_distr*.

Figure 81 and Figure 82 illustrate the energy flow for "*Condition 3* " and "*Condition 4*".

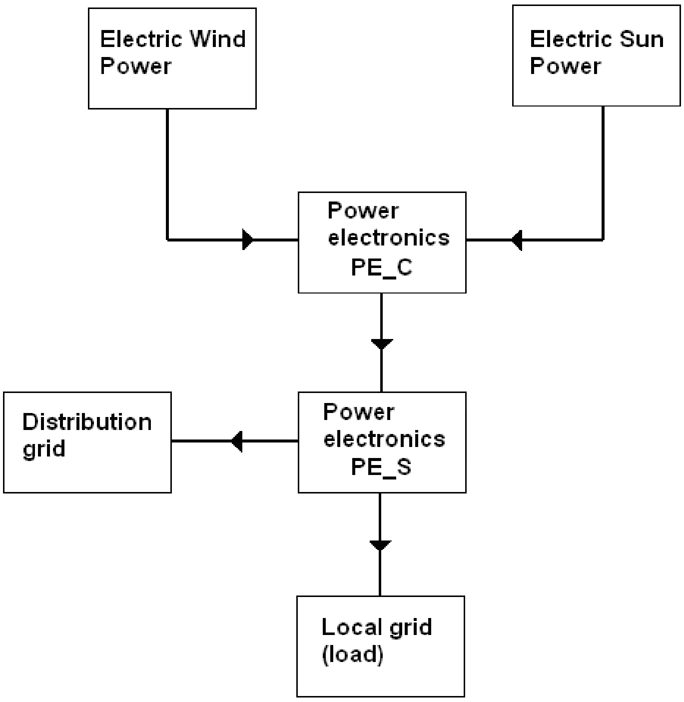

Figure 81 Energy flow for "*Condition 3* ". "Local grid" in the figure corresponds to "Consumer grid"

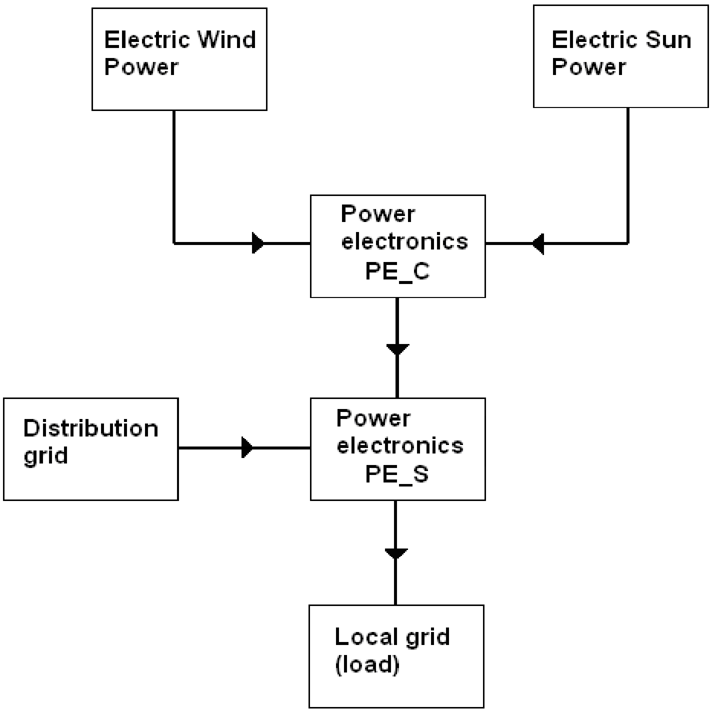

Figure 82 Energy flow for "*Condition 4*". "Local grid" in the figure corresponds to "Consumer grid"

4.8.2 Input parameters

The input parameters of the routine are specified in Table 37.

Parameter name	**Purpose**
P_buffer_vector (Input via function argument from Main Program)	See paragraph 4.7.1.2.
Sim_step_sec (Input via function argument from Main Program)	Time interval in seconds per simulation step (60 is a standard value)
charge_factor	Storage efficiency (part of unit) of the energy storage
load_factor	Loading efficiency (part of unit) of the energy storage
charge_level_init	Charge level in the energy storage at simulation start (kWh)
charge_level_min	Minimum allowed charge level in the energy storage (kWh)
charge_level_max	Maximum allowed charge level in the energy storage (kWh)
discharge_self	The energy storage self-discharge per 24 hours in % of charge_level_max
power_max_batt	Maximum power for storing / loading (kW) of the battery
power_max_distr	Maximum power for export / import (kW)

Table 37 Input parameters for routine “Battery_Distribution”

4.8.3 Output parameters

The output parameters of the routine are specified in Table 38.

Output is performed via function argument to Main Program.

Parameter name	Purpose
battery_charge_vector	Vector with current battery charge (per simulation step)
P_rel_batt_vector	Vector with the relation between Net Power (see See 4.7.1.2) to power_max_batt (see 4.8.2) (per simulation step)
P_rel_distr_vector	Vector with the relation between Net Power (see See 4.7.1.2) to power_max_distr (see 4.8.2) (per simulation step)
E_export_vector	Vector with export energy (per simulation step) (kWh)
E_import_vector	Vector with import energy (per simulation step) (kWh)
charge_level_max	See 4.8.2

Table 38 Output parameters for routine “Battery_Distribution”

4.8.4 Examples

Table 39 gives examples of used input parameters in a simulation.

Figure 83 shows the used input regarding Net Power vs time (P_buffer_vector).

Parameter name	Used value in the example
P_buffer_vector (Input via function argument from Main Program)	See Figure 83.
Sim_step_sec (Input via function argument from Main Program)	60
charge_factor	0.8
load_factor	0.9
charge_level_init	800
charge_level_min	600
charge_level_max	1000
discharge_self	0.5
power_max_batt	100
power_max_distr	400

Table 39 Examples of used input parameters for routine “Battery_Distribution”

The output results regarding *Battery charge, Net Power relative maximum storing / loading power, Net Power relative maximum export / import power, Imported Power and Exported Power* are shown in Figure 84 - Figure 88.

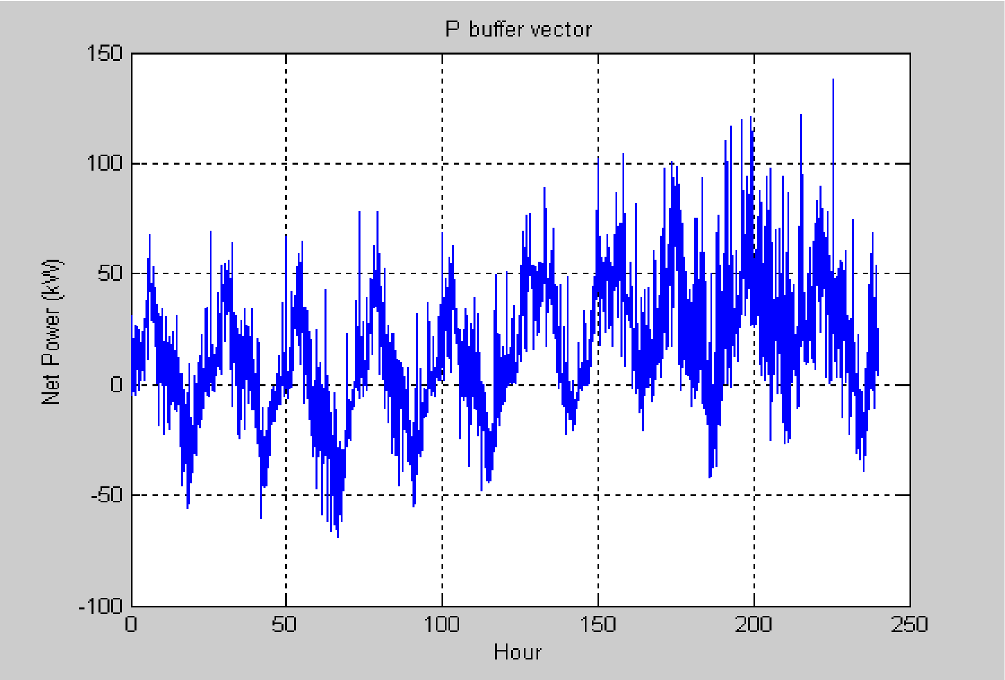

Figure 83 The Net Power vs Time (P_buffer_vector)

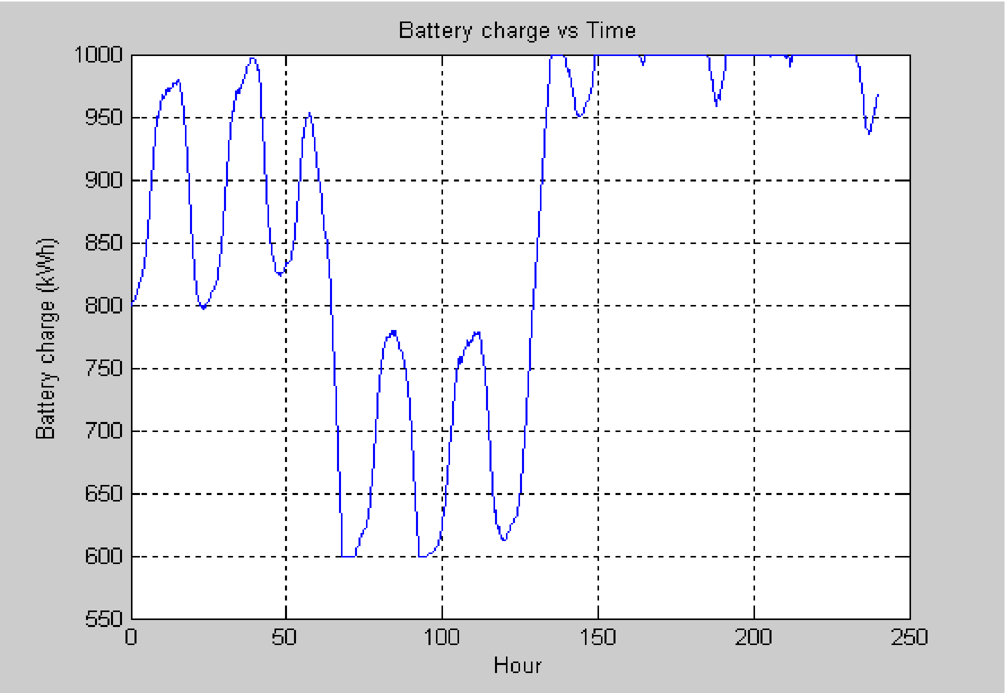

Figure 84 Battery charge vs Time

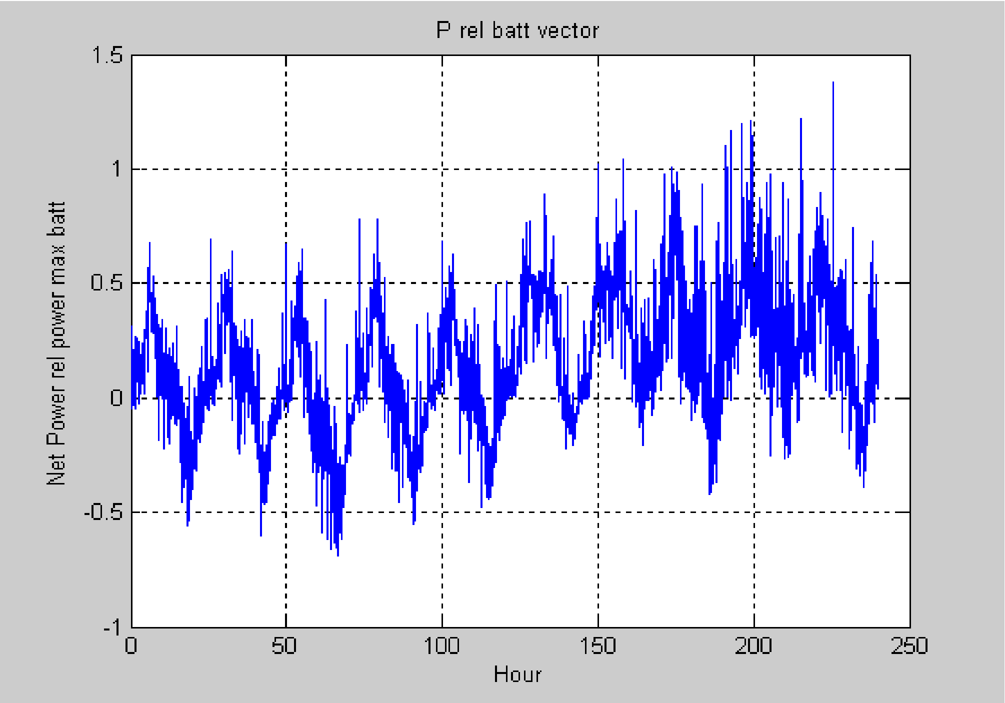

Figure 85 Net Power relative maximum storing / loading power vs Time

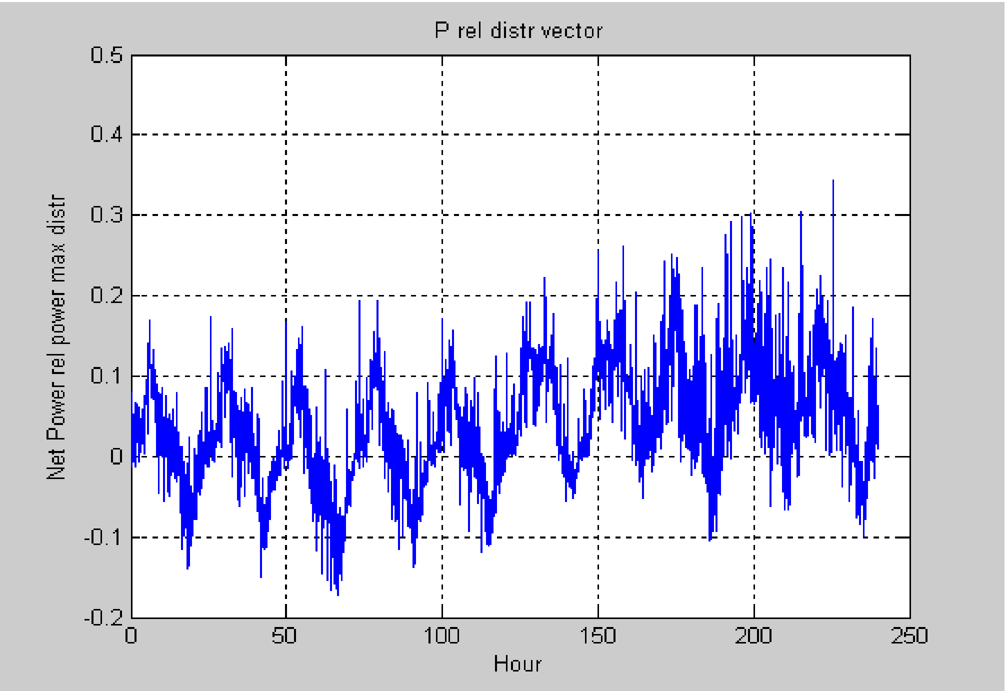

Figure 86 Net Power relative maximum export / import power vs Time

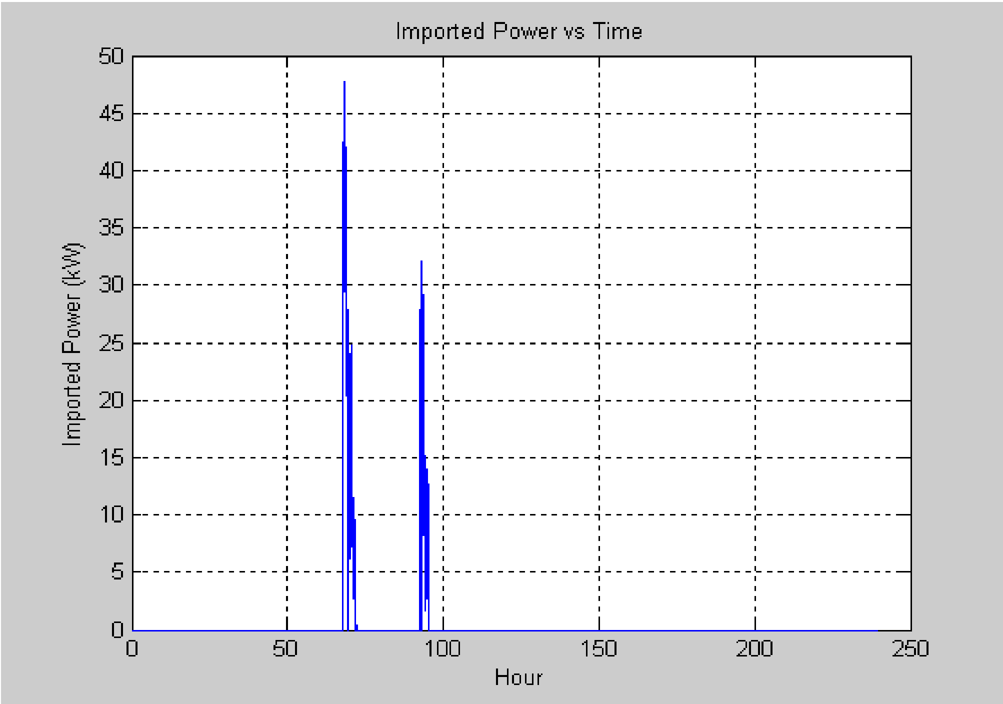

Figure 87 Imported Power vs Time

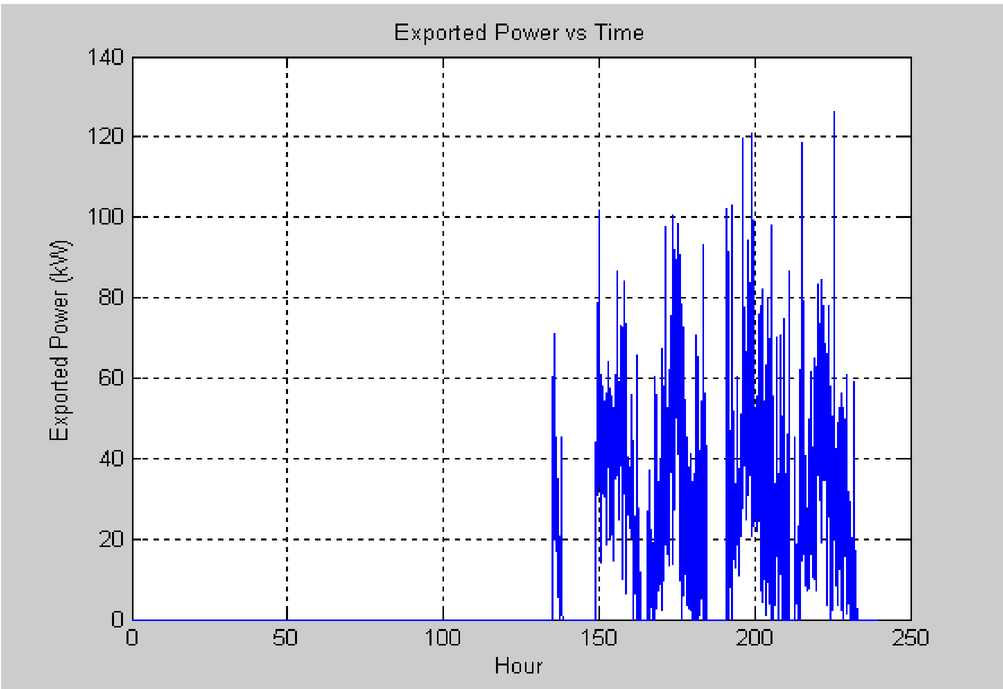

Figure 88 Exported Power vs Time

4.9 Power_evaluate

4.9.1 Principle

"Power_evaluate" is a program function (subroutine) that evaluates the result of a simulation process with a combined energy system that consists of the following building blocks:

- Wind power generators
- Sun power generators
- Loading grid (local grid)
- Energy storage (battery system)
- Distribution grid (for export / import of energy)

The routine presumes input data as a result of a number of repeating simulation sequences, where each sequence is a defined time period, e.g. a month. The total simulation process is a repetition, e.g. 100 times, of this sequence.

The following parameters are calculated (Table 40):

- Generated electric Wind Energy (*E_Wind*)
- Generated electric Sun Energy (*E_Sun*)
- Total generated (Wind + Sun) electric energy (*E_Gen*)
- Exported electric energy (*E_Exp*)
- Imported electric energy (*E_Imp*)
- Loaded (by the local grid) energy (*E_Load*)
- The difference between Exported and Imported electric energy (*D_Exp_Imp*)
- The quotient E_Wind / E_Gen (*Rel_Wind_Gen*)
- The quotient E_Sun / E_Gen (*Rel_Sun_Gen*)
- The quotient E_Gen / E_Load (*Rel_Gen_Load*)
- The quotient E_Exp / E_Gen (*Rel_Exp_Gen*)

• The quotient E_Imp / E_Load (*Rel_Imp_Load*) • The quotient D_Exp_Imp / E_Gen (*Rel_DEI_Gen*) • The quotient E_Battery / E_Load_day (*Rel_Battery_Load_day*). See comment below

Table 40 Parameters that are calculated

Comment regarding "The quotient E_Battery / E_Load_day"

E_Battery: maximum energy capacity of the battery system

E_Load_day: loaded (by the local grid) electric mean energy per 24 hours

All the above listed parameters all calculated according to:

- *per simulation sequence*. These values are stored in an *Evaluation file*. See Table 41, and Table 42.
- *mean value and standard deviation for the total simulation process*. These values are displayed after the simulation. As an example see paragraph 4.9.4

4.9.2 Input parameters

The input parameters of the routine are specified in Table 41.

Parameter name	**Purpose**
P_wind_el_vector (Input via function argument from Main Program)	Vector with electric wind power (per simulation step) (kW)
P_sun_el_vector (Input via function argument from Main Program)	Vector with electric sun power (per simulation step) (kW)
P_load_vector (Input via function argument from Main Program)	Vector with electric load (local) (per simulation step) (kW)
E_export_vector (Input via function argument from Main Program)	Vector with export energy (per simulation step) (kWh)
E_import_vector (Input via function argument from Main Program)	Vector with import energy (per simulation step) (kWh)

Sim_step_sec (Input via function argument from Main Program)	Time interval in seconds per simulation step (60 is a standard value)
charge_level_max (Input via function argument from Main Program)	Maximum allowed charge level in the energy storage (kWh)
N_sim_turns (Input via function argument from Main Program)	Number of simulation sequences in a total simulation
sim_turn (Input via function argument from Main Program)	Current simulation sequence
Evaluation_file	Name of an *evaluation file* (string) to store the parameters listed in Table 40 (per sequence) in vectors

Table 41 Input parameters for routine "Power_evaluate"

4.9.3 Output parameters

The output parameters of the routine are specified in Table 42.

Parameter name	**Purpose**
Evaluation_file	See Table 41

Table 42 Output parameters for routine "Power_evaluate"

4.9.4 Examples

The following list is an example of simulation results regarding mean values (My) and standard deviations (Sigma) based on 100 sequences:

E_Wind (kWh): My = 3.464e+003 Sigma = 1.230e+003

E_Sun (kWh): My = 4.204e+003 Sigma = 3.471e+002

E_Gen (kWh): My = 7.668e+003 Sigma = 1.294e+003

E_Load (kWh): My = 7.924e+003 Sigma = 3.831e+002

E_Exp (kWh): My = 6.455e+002 Sigma = 5.594e+002

E_Imp (kWh):	My = 1.517e+003	Sigma = 7.729e+002
D_Exp_Imp (kWh):	My = -8.713e+002	Sigma = 1.145e+003
Rel_Wind_Gen:	My = 4.371e-001	Sigma = 1.007e-001
Rel_Sun_Gen:	My = 5.629e-001	Sigma = 1.007e-001
Rel_Gen_Load:	My = 9.690e-001	Sigma = 1.662e-001
Rel_Exp_Gen:	My = 7.700e-002	Sigma = 6.076e-002
Rel_Imp_Load:	My = 1.906e-001	Sigma = 9.563e-002
Rel_DEI_Gen:	My = -1.432e-001	Sigma = 1.884e-001
Rel_Battery_Load:	My = 1.265e+000	Sigma = 6.117e-002

Figure 89 - Figure 98 give some information about the circumstances that have been present, during the simulation process. The figures in question illustrate only the different parameters for one single sequence (the first) but give a hint about the present situation.

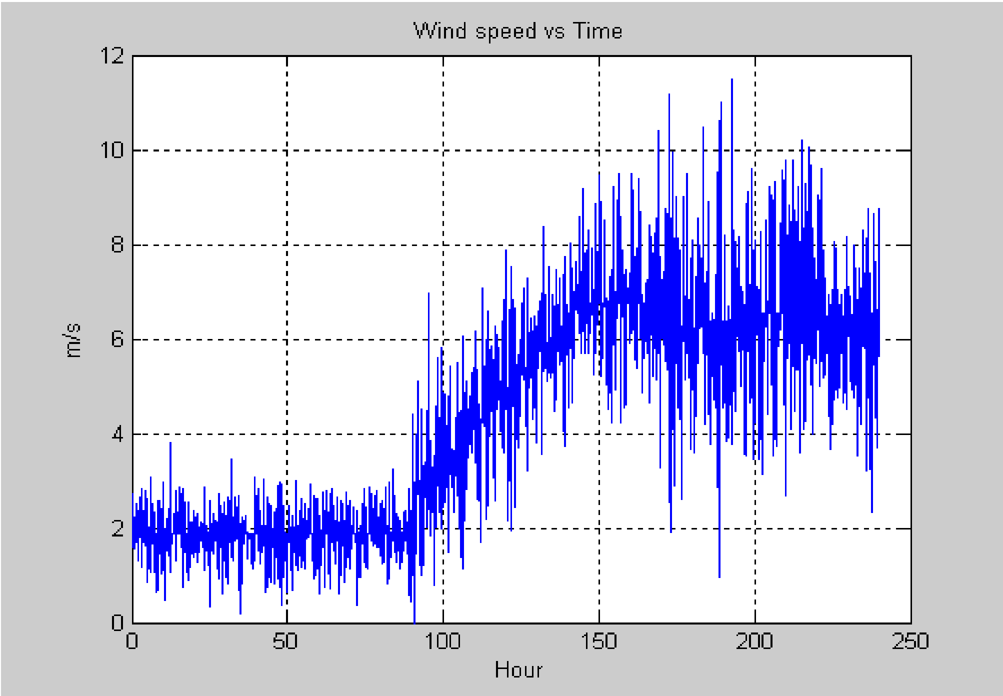

Figure 89 Wind speed vs Time

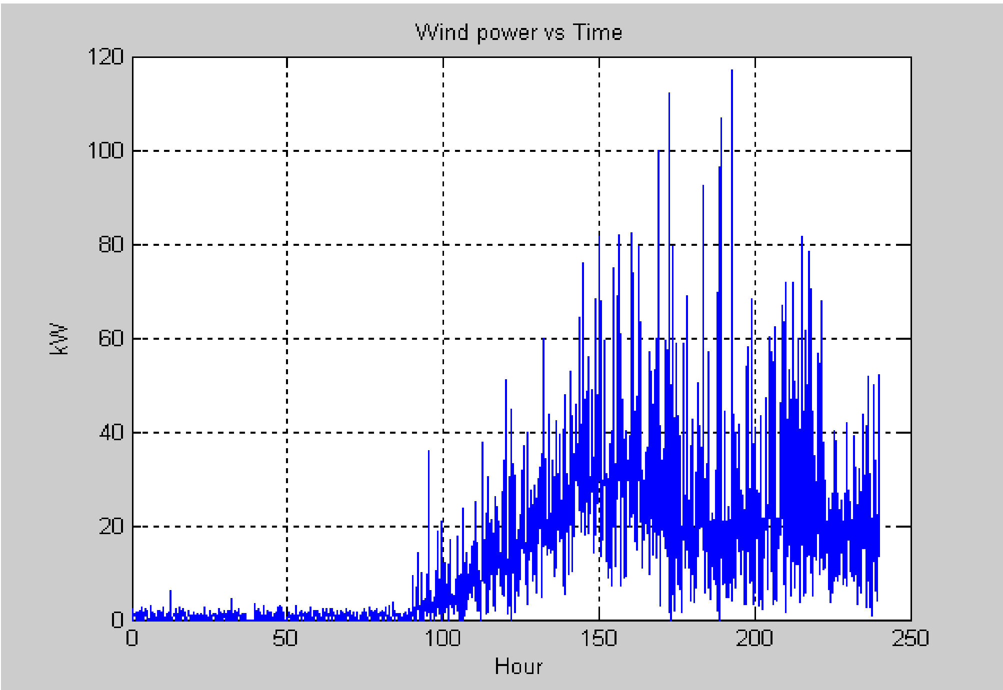

Figure 90 Wind power vs Time

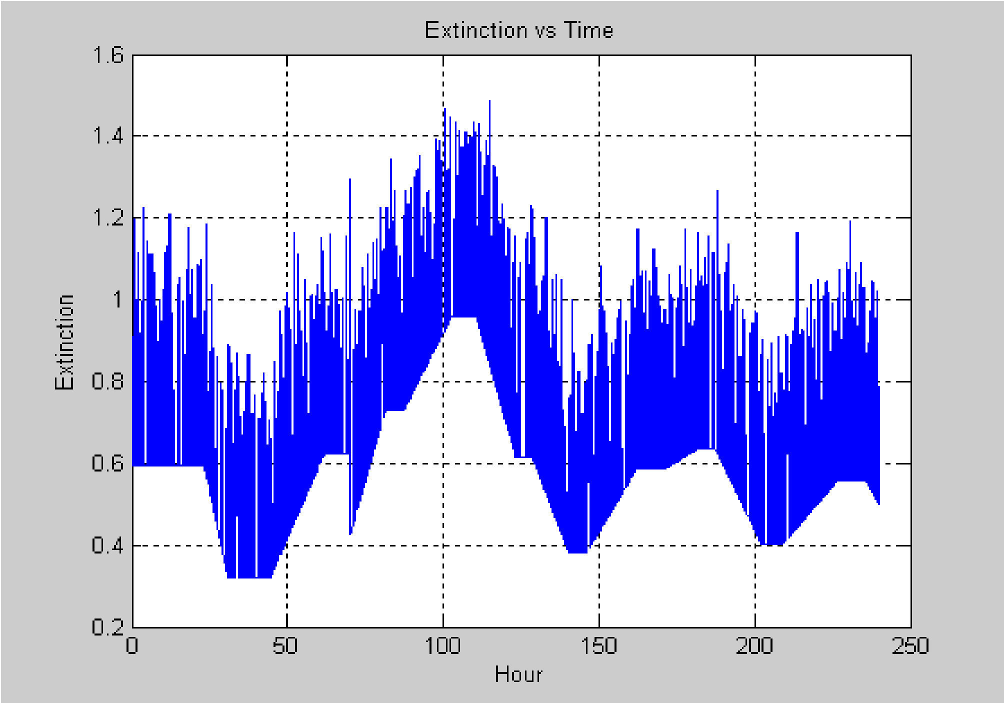

Figure 91 Extinction vs Time

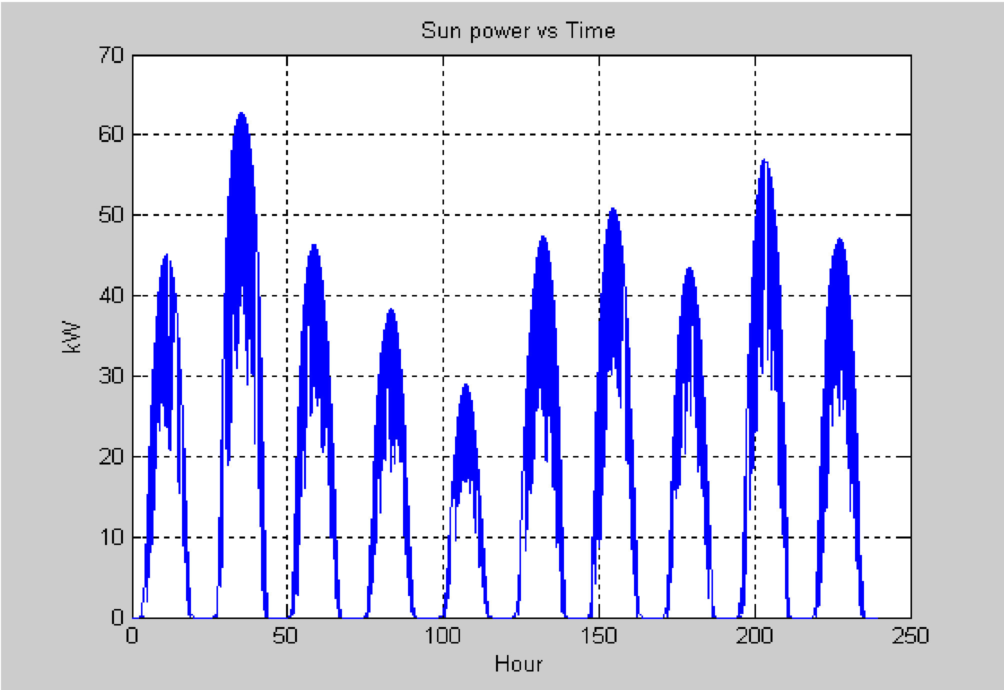

Figure 92 Sun power vs Time

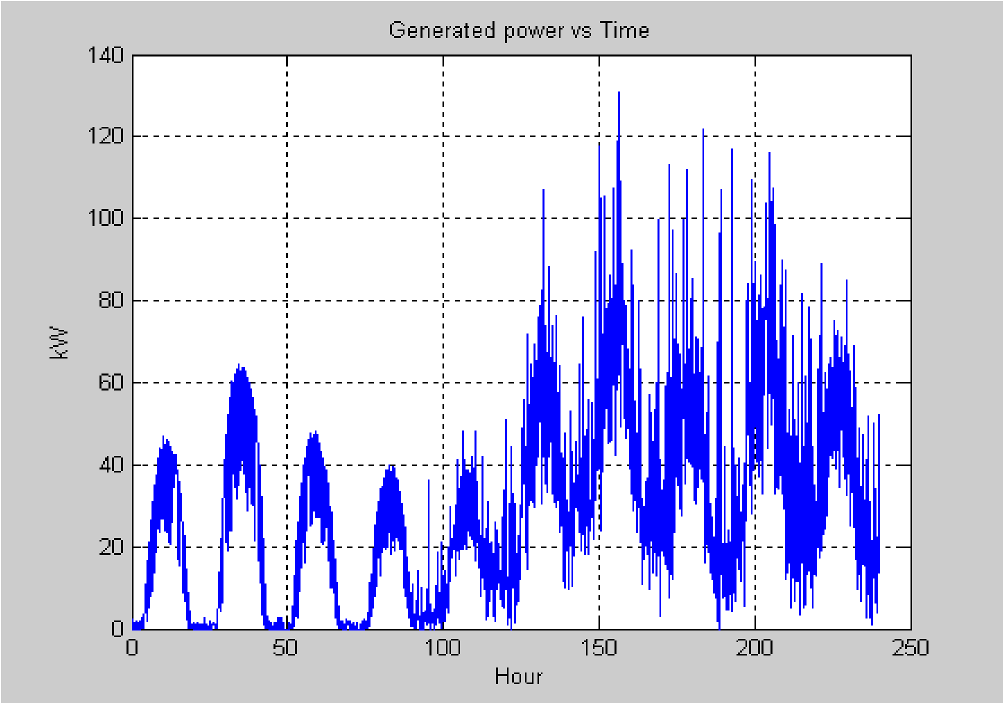

Figure 93 Gross power vs Time

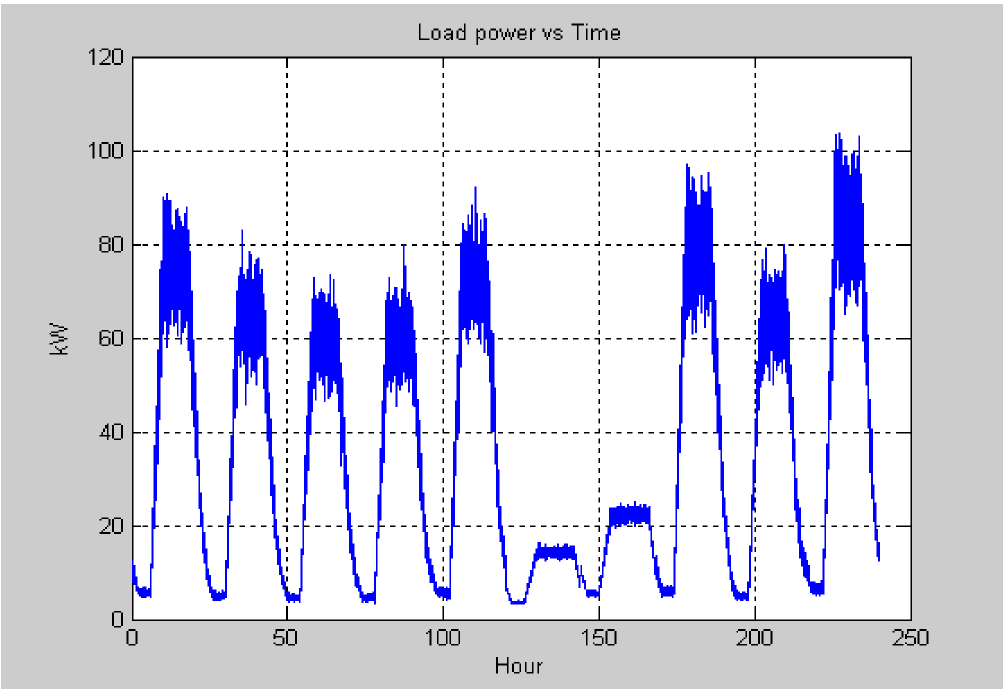

Figure 94 Load power vs Time

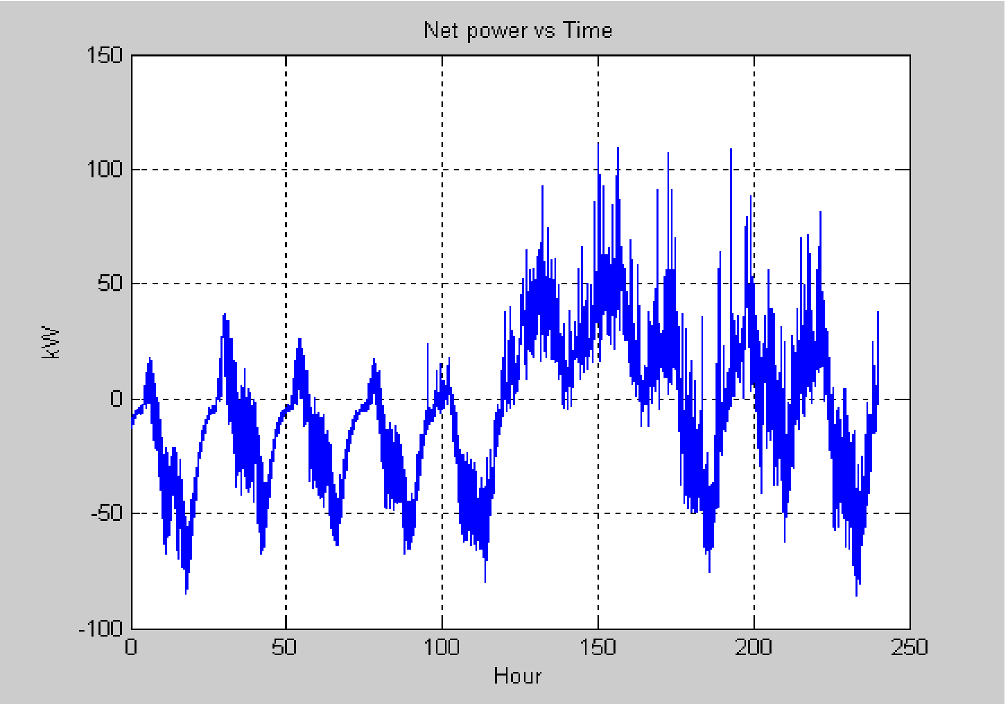

Figure 95 Net power vs Time

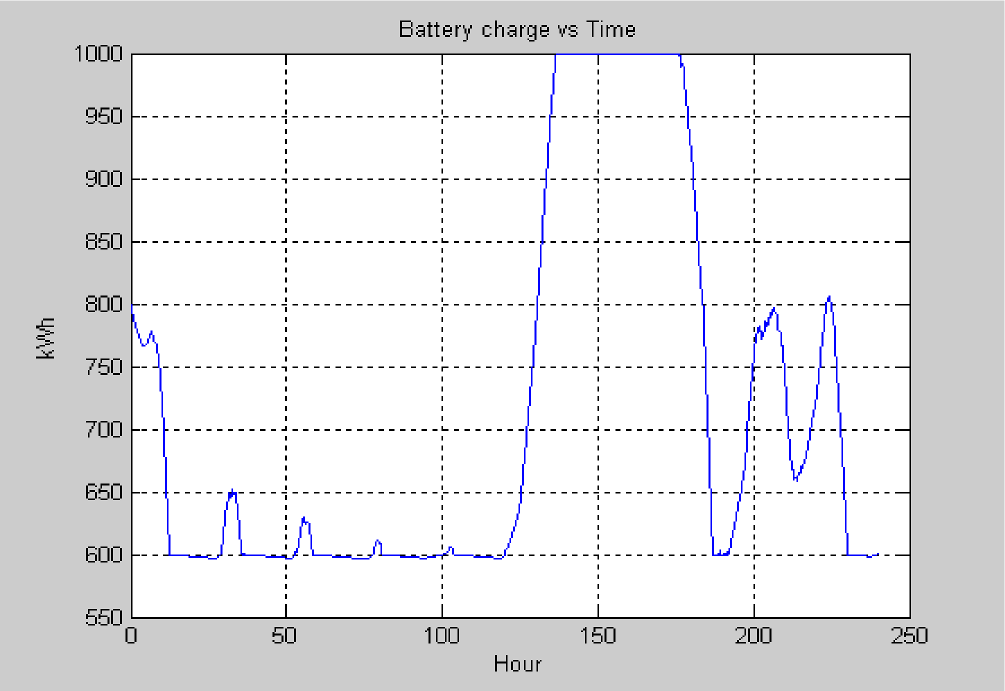

Figure 96 Battery charge vs Time

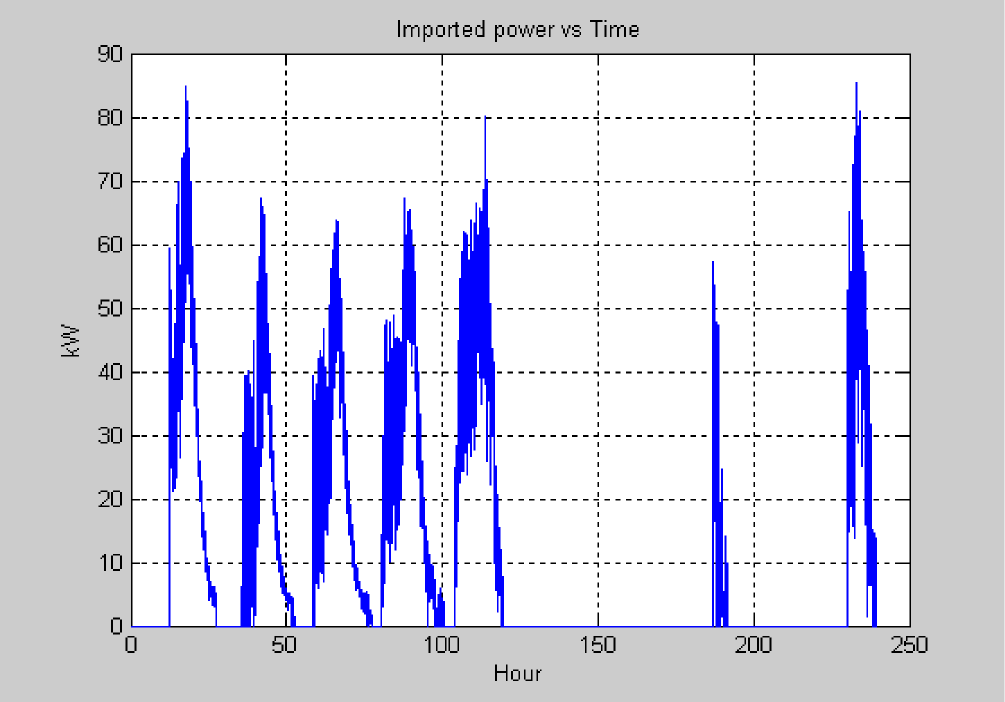

Figure 97 Imported power vs Time

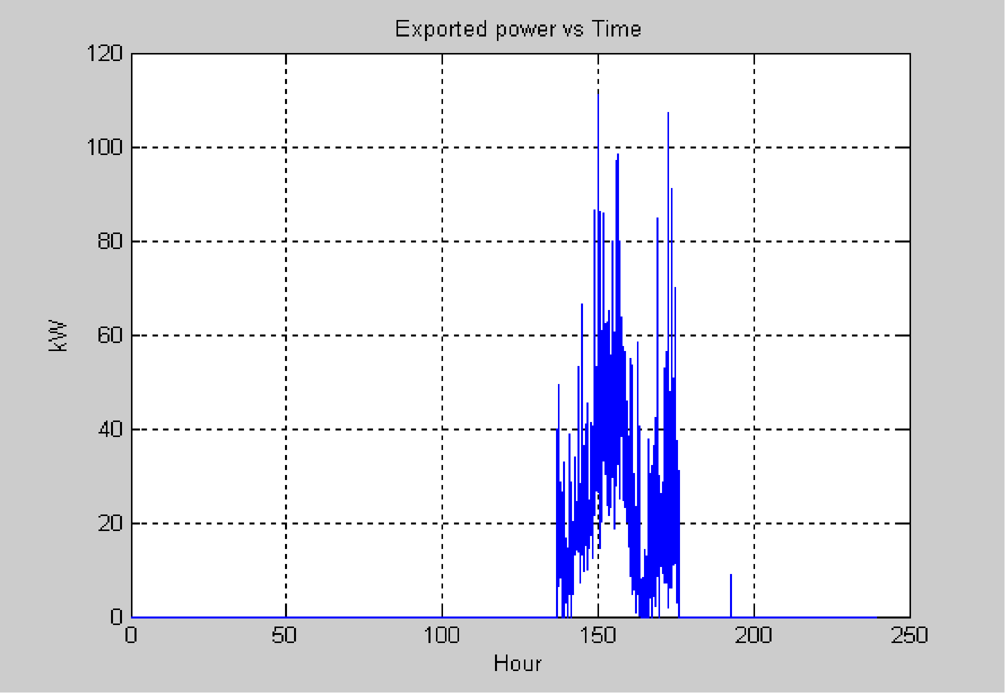

Figure 98 Exported power vs Time

5 REFERENCES

[1] Stochastic modeling of Wind Speed
Chalmers University of Technology, 2007
Ingemar Mathiasson

[2] Stochastic modeling of Extinction coefficents
Chalmers University of Technology, 2007
Ingemar Mathiasson

[3] Stochastic modeling of an Electrical Loads
Chalmers University of Technology, 2007
Ingemar Mathiasson

[4] Vind och Våggeneratorer
Chalmers University of Technology, 1998,
Ola Carlson

[5] Hand written notes regarding some measurements on Hönö Wind Turbine
Chalmers University of Technology,
Magnus Ellsén

[6] Algoritms to calculate planetary positions for Sun, Moon and major planets.
Developed by Paul Schlyter, based on T. van Flandern's and K. Pulkkinen's paper "Low precision formulae for planetary positions", published in the Astrophysical Journal Supplement Series, 1980