SDS

Solar Desalination Systems (SDS)

User Manual

Dr\ Mohamed. A. Sharafeldean, Faculty of Petroleum & Mining Engineering, Suez University, Egypt, 43722
msherafeldin@gmail.com, mohammed.eldeen@suezuniv.edu.eg

Contacts

- For any further information, please contact SDS developers at the following links:
  - msherafeldin@gmail.com
  - mohammed.eldeen@suezuniv.edu.eg
  - +60 1139795943 (mob)
  - Engineering science department, faculty of petroleum & mining engineering, Suez Canal University, Suez, Egypt.
  - PV21 condo, Setapak, Kuala Lumpur Malaysia, 53300
  - Contact persons: Dr. **Mohamed Abdel Wahab Sharaf**.

Contents

1. Introduction

Clean, fresh drinking water is essential to human and other life needs like agricultures needs, human needs and artificial needs. Access to safe drinking water has improved steadily and substantially over the last decades in almost every part of the world. However, some observers have estimated that by 2025 more than half of the world population will be facing water-based vulnerability, a situation which has been called a water crisis by the United Nations. Many remote areas of the world such as coastal desert areas in the Middle East or some Mediterranean are suffering from severe shortage of drinking water. Egypt, whose population of 80 million may reach 97 million by 2025, gets essentially no rainfall. All agriculture is irrigated by seasonal floods from the Nile River, and from water stored behind the Aswan High Dam. Egypt will face serious water shortages by the year 2025 and described the crisis as a possible drought. The rapid increase in population in Egypt, moreover miscarriage, absence of good management of water distribution and water pollution will be expected as the main reasons for the shortage. Exploitation of natural fresh water resources combined with higher water demand has led to an increased demand for alternative fresh water resources.

Desalination provides such an alternative source, offering water otherwise not accessible for irrigational, industrial and municipal use. Desalination technologies can be classified by their separation mechanism into thermal and membrane based desalination. Thermal desalination separates salt from water by evaporation and condensation, whereas in membrane desalination water diffuses through a membrane, while salts are almost completely retained. Thermal desalination includes multi-stage flash, multi-effect distillation, mechanical vapor compression, and thermal vapor compression and membrane desalination contains reverse osmosis, ion exchange, and electro-dialysis processes. Reverse osmosis and multi-stage flash are the techniques that are most widely used. The decision for a certain desalination technology is influenced by feed water salinity, required product quality as well as by site-specific factors such as labor cost, available area, energy cost and local demand for electricity.

Desalination uses a large amount of energy to remove a portion of pure water from a salt water source. It has been estimated that the production of 1000 m^3 per day of freshwater requires 10,000 tons of oil per year. This is highly significant as it involves a recurrent energy expense which few of the water-short areas of the world can afford. Large commercial desalination plants using fossil fuel are in use in a number of oil-rich countries to supplement the traditional sources of water supply. People in many other areas of the world have neither the money nor oil resources to allow them to develop on a similar manner. Problems relevant to the use of fossil fuels, in part, could be resolved by considering possible utilization of renewable resources such as solar, biomass, wind, or geothermal energy. It often happens that the geographical areas where water is needed are well gifted with renewable energy sources. Thus, the obvious way is to combine those renewable energy sources to a desalination plant, in order to provide water resources as required.

Recently, considerable attention has been given to the use of renewable energy as sources for desalination, especially in remote areas and islands, because of the high costs of fossil fuels, difficulties in obtaining it, attempts to conserve fossil fuels, interest in reducing air pollution, and the lack of electrical power in remote areas. It is, however, to be noted that in spite of the aforesaid favorable characteristics, the renewable energy contribution to cover energy demand worldwide, though increasing, is still marginal. Aside from the hydroelectric energy, the other principal resources (solar, wind, geothermal) cover together little more than 1% of the energy production worldwide. Owing to the diffuse nature of solar energy, the main problems with the use of solar thermal energy in large-scale desalination plants are the relatively low productivity rate, the low thermal efficiency and the considerable land area required. However, since solar desalination plants are characterized by free energy and insignificant operation cost, this technology is, on the other hand, suitable for small-scale production, especially in remote arid areas and islands, where the supply of conventional energy is scarce. The coupling of renewable energy sources with desalination processes is seen by some as having the potential to offer a sustainable route for increasing the supplies of potable water. Solar energy can directly or indirectly be harnessed for desalination. Collection systems that use solar energy to produce distillate directly in the solar collector are called direct collection systems whereas systems that combine solar energy collection systems with conventional desalination systems are called indirect systems. In indirect systems, solar energy is used either to generate the heat required for desalination and/or to generate electricity that is used to provide the required electric power for conventional desalination plants such as multi-effect (ME), multi-stage flash (MSF) or reverse osmosis (RO) systems. For Middle East countries sun has a good presence beside a huge area of the desert. For example the number of sun shine hours over Egypt is about 3600hr/year.

Therefore utilization of solar energy as an alternative and renewable energy should be strongly taken into consideration, especially, when new communities are established in the desert and remote areas. Therefore, it is possible to use this energy to produce fresh water from the sea or underground saline water (i.e., solar desalination) especially for remote areas where a conventional energy supply is not easily available. This application is, however, still not well developed and it has only been tested in pilot plants and at a few demonstration sites. For scientists, it is very important to decide or demonstrate the applicability of the solar desalination system based on energy, exergy, cost, and thermo-economic analysis. And the decision should maintain different types, different configurations, and different techniques. So to perform a reliable analysis for this wide range of solar desalination processes and different configurations, a flexible visualized computer program is required.

The software program should perform different types of calculations (performance, design), different analysis (energy, exergy, cost, thermo-economic), and different comparisons and modifications. Therefore, the need to design and simulate the global solar desalination systems is very important and essential. The main objective is to develop software to simulate, design, and optimize different solar desalination systems such as Reverse Osmosis, Multi stage Flash, Multi Effect Evaporation, Mechanical and Thermal Vapor Compression. The developed software should perform different calculations, different modifications, different comparisons, and different analysis. Optimization process will be embedded and considered. The developed software should have validity, generality, flexibility, easy to handle, and executable. Programming with MatLab\SimuLink tackles the power of the streams leading to flexibility, generality, and ease of programming. It is become easy to define more than one input to get one or more outputs.

Therefore, SimuLink considered a new generation software tool to design what would be desired by the designers. A Solar desalination systems (SDS) program is developed for design and simulation of different types and configurations of solar desalination processes. Embedded block programming with MatLab\SimuLink is used to offer a flexible reliable and friendly user-interface. The desalination plant components such as heat exchangers, flash chambers, evaporators, pumps, steam ejectors, compressors, reverse osmosis membranes, pipes, etc., are stored as blocks in SimuLink visual library, which enables the user to construct different configurations by just clicking the mouse over the required units (blocks). The interface aids plant designers, operators and other users to perform different calculations such as energy, exergy, and thermo-economics. In addition, the package enables the designers to perform different modifications of an existing plant or to develop the conceptual design of new configurations. Some operating desalination plants are simulated by the present package to show its reliability and flexibility. SDS package have some features concluded in:

- Easy model construction.
- Easy to convert the designed code to be self executable and work under different computer languages (Visual basic, Visual C, Visual C++, and Visual Fortran).
- The model allows users to easily change the plant variables and different operating conditions with ultimate stream allowance.
- The developed program overcomes the problem appears in other techniques of simulation such as object oriented approach, sequential approach, matrix manipulation technique. The use of SimuLink overcomes the power of streams in the simulation program without tearing or removing some streams.

2. Working fluids: thermo physical properties

2.1 Water thermo physical properties

2.1.1 Density kg/m^3

The equation is applicable in the temperature range of 10 to 180°C and for salinity from 0 to 160 g/kg.

$$\rho = 0.5 \times a_o + a_1 \times Y + a_2 \times (2 \times Y^2 - 1) + a_3 \times (4 \times Y^3 - 3 \times Y)$$

Where:

$$a_o = 2.016110 + 0.115313 \times \sigma + 0.000326 \times (2 \times \sigma^2 - 1)$$

$$a_1 = -0.0541 + 0.01571 \times \sigma - 0.000423 \times (2 \times \sigma^2 - 1)$$

$$a_2 = -0.006124 + 0.00174 \times \sigma - 0.000009 \times (2 \times \sigma^2 - 1)$$

$$a_3 = 0.000346 + 0.000087 \times \sigma - 0.000053 \times (2 \times \sigma^2 - 1)$$

$$Y = \frac{2t - 200}{160}$$

$$\sigma = \frac{2X - 150}{150}$$

2.1.2 Dynamic viscosity

The Validity range of this correlation is 10-150 °C and 0-130 g/kg salt concentration.

$$\mu = \mu_w \times \mu_R$$

μ_w: Viscosity of pure water.

μ_R: Relative viscosity and =1 for pure water & > 1 for salt solution.

$$\ln \mu_w = -3.79418 + \frac{604.129}{139.18 + t}$$

t: Temperature in °C.

$$\mu_R = 1 + a_1 \times X + a_2 \times X^2$$

Where,

$$a_1 = 1.474 \times 10^{-3} + 1.5 \times 10^{-5} \times t - 3.927 \times 10^{-8} \times t^2$$

$$a_1 = 1.0734 \times 10^{-5} + 8.5 \times 10^{-8} \times t - 2.23 \times 10^{-10} \times t^2$$

2.1.3 Boiling point elevation

This equation is valid for X (salt concentration) from 20 to 160 g/kg for t (temperature) from 20 to 180°C.

$$BPR = (B + C \times X) \times X$$

Where:

$$10^3 \times B = 6.71 + 6.43 \times 10^{-2} \times t + 9.74 \times 10^{-5} \times t^2$$

$$10^5 \times C = 2.38 + 9.59 \times 10^{-3} \times t + 9.42 \times 10^{-5} \times t^2$$

2.1.4 Specific heat capacity kJ/kg°C

$$C_p = A + B\times t + C\times t^2 + D\times t^3$$

Where:

$$A = 4206.8 - 6.6197\times X + 1.2288\times 10^{-2}\times X^2$$

$$B = -1.1262 + 5.4178\times 10^{-2}\times X - 2.2719\times 10^{-4}\times X^2$$

$$C = 1.2026\times 10^{-2} - 5.3566\times 10^{-4}\times X + 1.8906\times 10^{-6}\times X^2$$

$$D = 6.8774\times 10^{-7} + 1.517\times 10^{-6}\times X - 4.4268\times 10^{-9}\times X^2$$

X is the salinity ratio in kg/kg

2.1.5 Thermal conductivity W/m°C

$$k = A + B\times t + C\times t^2$$

$$A = 576.6 - 34.64\times CA + 7.286\times CA^2$$

$$B = 10^{-3}\times(1526 + 466.2\times CA - 226.8\times CA^2 + 28.67\times CA^3)$$

$$C = -10^{-5}\times(581 + 2055\times CA - 991.6\times CA^2 + 146.4\times CA^3)$$

And,

$$CA = \frac{28.17\times X}{1000 - X}$$

2.1.6 Latent heat of vaporization kJ/kg

$$L = 2501.897149 - 2.407064037\times \mathrm{T} + 1.192217\times 10^{-3}\times \mathrm{T}^2 - 1.5863\times 10^{-5}\times \mathrm{T}^3$$

2.1.7 Saturation pressure bar

$$P_{sat} = 872.3\times \exp^{-(\mathrm{T}-585.5/169.5)^2} + 39.07\times \exp^{-(\mathrm{T}-342.4/124.4)^2}$$

2.1.8 Specific enthalpy of dry saturated vapor kJ/kg

$$h_v = -3.078\mathrm{e} - 18\times \mathrm{T}^9 + 4.762\mathrm{e} - 15\times \mathrm{T}^8 - 3.076\mathrm{e} - 12\times \mathrm{T}^7 + 1.074\mathrm{e} - 9\times \mathrm{T}^6 - 2.193\mathrm{e} - 7\times \mathrm{T}^5 + 2.646\mathrm{e} - 5\times \mathrm{T}^4 - 0.001824\times \mathrm{T}^3 + 0.06417\times \mathrm{T}^2 + 0.894\times \mathrm{T} + 2504$$

2.1.9 Specific enthalpy of saturated liquid kJ/kg

$$h_l = -0.033635409 + 4.207557011\times \mathrm{T} - 6.200339\times 10^{-4}\times \mathrm{T}^2 + 4.459374\times 10^{-6}\times \mathrm{T}^3$$

2.1.10 Specific entropy of saturated liquid kJ/kg°C

$$s_l = -0.00057846 + 0.015297489\times \mathrm{T} - 2.63129\times 10^{-5}\times \mathrm{T}^2 + 4.11959\times 10^{-8}\times \mathrm{T}^3$$

2.1.11 Specific entropy of saturated vapor kJ/kg°C

$$s_v = -4.697\mathrm{e}-21 \times \mathrm{T}^9 + 7.274\mathrm{e}-18 \times \mathrm{T}^8 - 4.706\mathrm{e}-15 \times \mathrm{T}^7 + 1.648\mathrm{e}-12 \times \mathrm{T}^6 - 3.39\mathrm{e}-10 \times \mathrm{T}^5 + 4.2\mathrm{e}-8 \times \mathrm{T}^4 - 3.27\mathrm{e}-6 \times \mathrm{T}^3 + 0.0002231 \times \mathrm{T}^2 - 0.02823 \times \mathrm{T} + 9.16$$

2.2 Toluene thermo physical properties

2.2.1 Density for liquid and vapor phases kg/m³

$$\rho_{tl} = -7.981^{-19} \times T^9 + 7.002^{-16} \times T^8 - 2.087^{-13} \times T^7 + 1.821^{-11} \times T^6 + 1.971^{-9} \times T^5 - 3.474^{-7} \ldots \times T^4 - 3.29^{-6} \times T^3 + 0.001316 \times T^2 - 0.9326 \times T + 884.5$$

$$\rho_{tv} = 7.873^{15} \times exp^{-((Tco-868.2)/97.11)^2)} + 1898 \times exp^{-((Tco-666.7)/219.2)^2)}$$

2.2.2 Dynamic viscosity for liquid and vapor phases kg/m³

Toluene physical properties for liquid and vapor phases are obtained from the following correlations;

$$\mu_{tl} = 10^{-6} \times (3.262729 \times 10^{-5} \times T^3 + 5.14015 \times 10^{-2} \times T^2 - 27.89675 \times T + 5.305598 \times 10^3)$$

$$\mu_{tv} = 10^{-6} \times (6.338982 \times 10^{-8} \times T^4 - 1.602562 \times 10^{-4} \times T^3 + 1.519286 \times 10^{-1} \times T^2 \ldots - 63.99838 \times T + 1.011961 \times 10^4)$$

2.2.3 Specific enthalpy of dry saturated vapor kJ/kg

$$h_v = 2.323\mathrm{e}-019 \times \mathrm{T}^9 + 2.638\mathrm{e}-16 \times \mathrm{T}^8 - 7.835\mathrm{e}-14 \times \mathrm{T}^7 + 6.784\mathrm{e}-12 \times \mathrm{T}^6 + 7.627\mathrm{e}-10 \times \mathrm{T}^5 - 1.392\mathrm{e}-7 \times \mathrm{T}^4 - 1.443\mathrm{e}-6 \times \mathrm{T}^3 + 0.002331 \times \mathrm{T}^2 + 1.019 \times \mathrm{T} + 490.4$$

2.2.4 Specific enthalpy of saturated liquid kJ/kg

$$h_l = -3.023\mathrm{e}-19 \times \mathrm{T}^9 - 2.041\mathrm{e}-16 \times \mathrm{T}^8 + 6.098\mathrm{e}-14 \times \mathrm{T}^7 - 5.372\mathrm{e}-12 \times \mathrm{T}^6 - 5.526\mathrm{e}-10 \times \mathrm{T}^5 + 9.276\mathrm{e}-8 \times \mathrm{T}^4 + 2.962\mathrm{e}-6 \times \mathrm{T}^3 + 0.001018 \times \mathrm{T}^2 + 1.628 \times \mathrm{T} + 63.19$$

2.2.5 Specific entropy of saturated vapor kJ/kg°C

$$s_l = -6.571\mathrm{e}-16 \times \mathrm{T}^6 - 7.761\mathrm{e}-14 \times \mathrm{T}^5 + 2.712\mathrm{e}-10 \times \mathrm{T}^4 - 1.128\mathrm{e}-7 \times \mathrm{T}^3 + 2.61\mathrm{e}-5 \times \mathrm{T}^2 - 0.001973 \times \mathrm{T} + 1.813$$

2.2.6 Specific entropy of saturated liquid kJ/kg°C

$$s_l = 1.038 \times exp^{(0.002218 \times T)} - 0.7889 \times exp^{(-0.004717 \times T)}$$

2.2.7 Saturation pressure bar

$$P_{sat} = 7.025e-22 \times T^9 - 4.53e-19 \times T^8 + 1.187e-16 \times T^7 - 2.775e-14 \times T^6 + 6.104e-12 \times T^5 + 2.474e-9 \times T^4 + 2.434e-7 \times T^3 + 1.429e-5 \times T^2 + 0.0005795 \times T + 0.009935$$

2.3 Therminol-VP1 heat transfer oil thermo physical properties

2.3.1 Specific heat capacity kJ/kg°C

$$C_p = -0.6622 \times exp^{(0.001186 \times T)} + 2.178 \times exp^{(0.0007637 \times T)}$$

2.3.2 Pressure bar

$$P = 1.059e - 9 \times T^4 - 3.412e - 7 \times T^3 + 3.867e - 5 \times T^2 - 0.001491 \times T + 0.01249$$

2.3.3 Specific enthalpy kJ/kg

$$h = 0.00137 \times T^2 + 1.5 \times T - 18.46$$

2.3.4 Specific entropy kJ/kg°C

$$s = 1.038 \times exp^{(0.002218 \times T)} - 0.7889 \times exp^{(-0.004717 \times T)}$$

For further information about the related working fluids, please visit the following links:

- www.therminol.com
- http://webbook.nist.gov/chemistry/

3. Solar desalination units: Processes Mathematical models

The demanded power from desalination plants (block) permit the use of medium size of power source from solar field in case of direct vapor generation technique. Therefore, solar desalination plants (SDP) often contains the following:

- Low and/or medium temperature solar collectors (FPC, CPC, PTC).
- Solar field recirculation pumps.
- Control valves, sensors, flow meters, tanks, regulators, …
- Organic Rankine cycle (turbine, condenser, heat exchanger, recuperator …) in case of producing electricity for mechanical parts.
- Desalination blocks (thermal, membrane and/or mechanical) depending on the technique and the supplying method.

The application of solar energy to produce fresh water is receiving increased interest due to the need for solving the water shortage problems in various areas of the world at the same time as conventional energy sources used for obtaining water in different scenarios become depleted. Over the past few decades, the reverse osmosis (RO) process of seawater desalination has gained much popularity. RO is a membrane process, and was developed in direct competition with distillation processes. Its main feature is that it requires no thermal energy but, rather, mechanical energy in the form of a high pressure pump. Solar thermal energy coupled to a power cycle by using direct mechanical power can also be employed. Solar troughs and linear Fresnel can concentrate the sunlight by about **70–100 times**. Typical operating temperatures are in the range of **350–550°C**. Plants of 200MW rated power and more can be built by this technology.

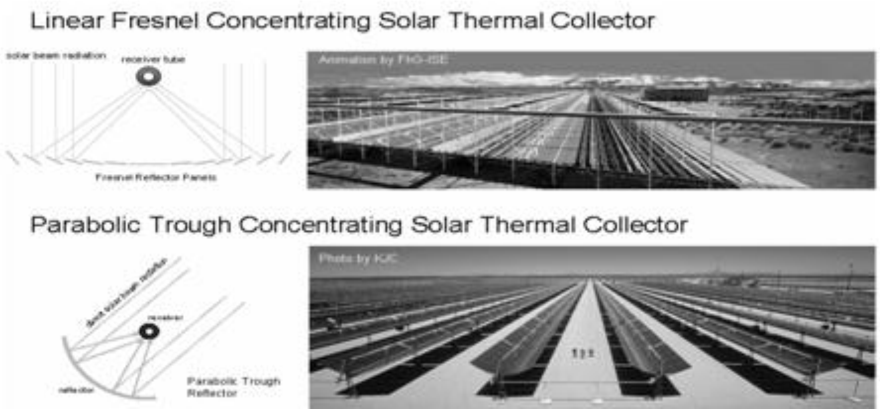

Linear Fresnel Concentrating Solar Thermal Collector
solar beam radiation
receiver tube
Fresnel Reflector Panels
Animation by FhG-ISE
Parabolic Trough Concentrating Solar Thermal Collector
direct solar beam radiation
receiver
reflector
Parabolic Trough Reflector
Photo by KJC

Figure (1) Linear and Parabolic troughs for solar concentrated thermal power.

The CSP technologies have some features:

- Concentrating solar power plants can generate electricity which can be used for membrane desalination.
- CSP plants can be used for combined heat and power.
- Thermal desalination methods like multi effect distillation (MED) or multi stage flash (MSF) can be powered by CSP, either directly or in co-generation with electricity.
- CSP reduces emissions of local pollutants and considerably contribute to global climate protection.

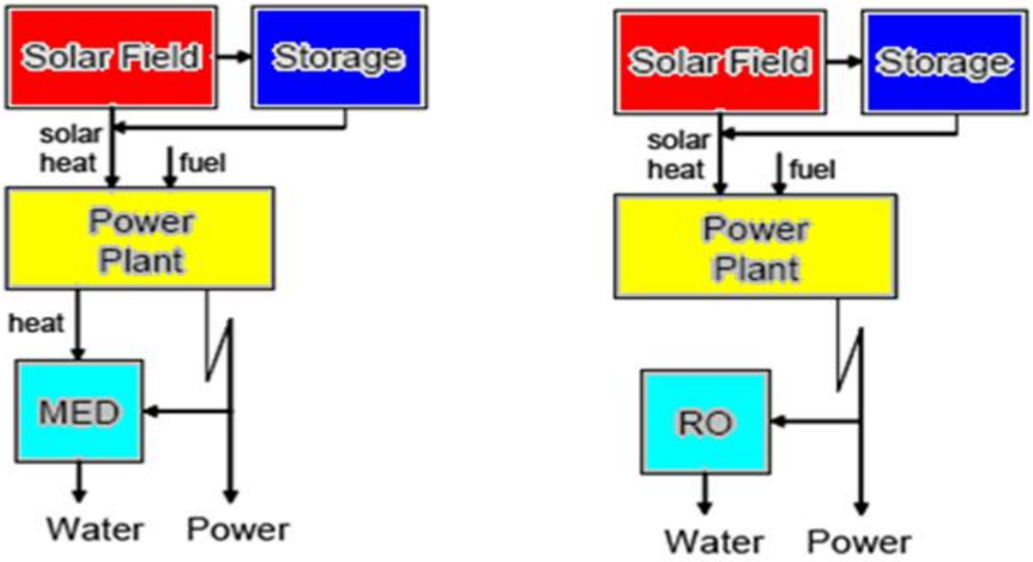


Figure (2) CSP's technology implementation for different types of thermal desalination plants.

In this part, it is very important to highlight on the different mathematical models for different processes of solar desalination plants. Solar radiation model, different types of solar collectors, different types of energy units, different types of desalination plants are pinpointed and mathematically analyzed based on approaches such as energy, exergy, thermo-economic, and cost.

3.1 Mathematical approaches

The mathematical approaches that used in the analysis for solar desalination plants are basically preformed according to 1st and 2nd laws of thermodynamics. For any system goes under steady state, the mass, energy, and entropy balances equations under steady state condition should be developed as following;

$$\sum \dot{m}_{in} - \sum \dot{m}_{out} = 0, kg/s$$

$$\sum \dot{e}_{in} - \sum \dot{e}_{out} = 0, kJ/kg$$

$$\sum \dot{s}_{in} - \sum \dot{s}_{out} = 0, kJ/kgoC$$

Unlike energy, which is conserved in any process according to the first law of thermodynamics, exergy is destroyed due to irreversibility taking place in any process, which manifests itself in entropy creation or entropy increase. The general form of the availability is defined by the following equation;

$$A_2 - A_1 = A_q + A_w + A_{fi} - A_{fo} - I$$

Where A_2-A_1=0 is the non-flow availability change in steady state condition, $A_q = \sum_J (1 - T_{amb}/T_J) Q_J$ is the availability transfer due to the heat transfer between the control volume and its surroundings, $A_w = -W_{cv} + P_o(V_2 - V_1)$ is equal to the negative value of the work produced by the control volume but in most cases the control volume has a constant volume, therefore A_w can be further simplified. And I=T_{amb}×S_{gen} is the availability destruction in the process. The flow availability expressed as$A_{fi,o} = \sum_{i,o} \dot{m}_{i.o}\, a_{fi,o}$. So the general form in steady state condition would become;

$$0 = A_q + A_w + A_{fi} - A_{fo} - I$$

Thermo-economic is the branch of engineering that combines exergy analysis and economic principles to provide the system designer or operator with information not available through conventional energy analysis and economic evaluations but crucial to the design and operation of a cost effective system. In a conventional economic analysis, a cost balance is usually formulated for the overall system operating at steady state as following;

$$\sum_{out} C^{\cdot} = \sum_{in} C^{\cdot} + Z^{IC\&OM}$$

Where $C^{\cdot}$ the cost rate according to inlet and outlet streams, and $Z^{IC\&OM}$ is the capital investment and operating & maintenance costs. In exergy costing a cost is associated with each exergy stream. Thus, for inlet and outlet streams of matter with associated rates of exergy transfer $E_{i,o}$, power W, and the exergy transfer rate associated with heat transfer E_q it can write as following;

$$C_{\dot{i},o} = c_{i,o}E_{\dot{i},o}$$
$$C_{\dot{w}} = c_w W^{\cdot}$$
$$C_{\dot{q}} = c_q E_{\dot{q}}$$

Where $c_{i,o,w,q}$ denote average costs per unit of exergy in \$/kJ for inlet ($_i$), outlet ($_o$), power ($_w$), and energy ($_q$) respectively.

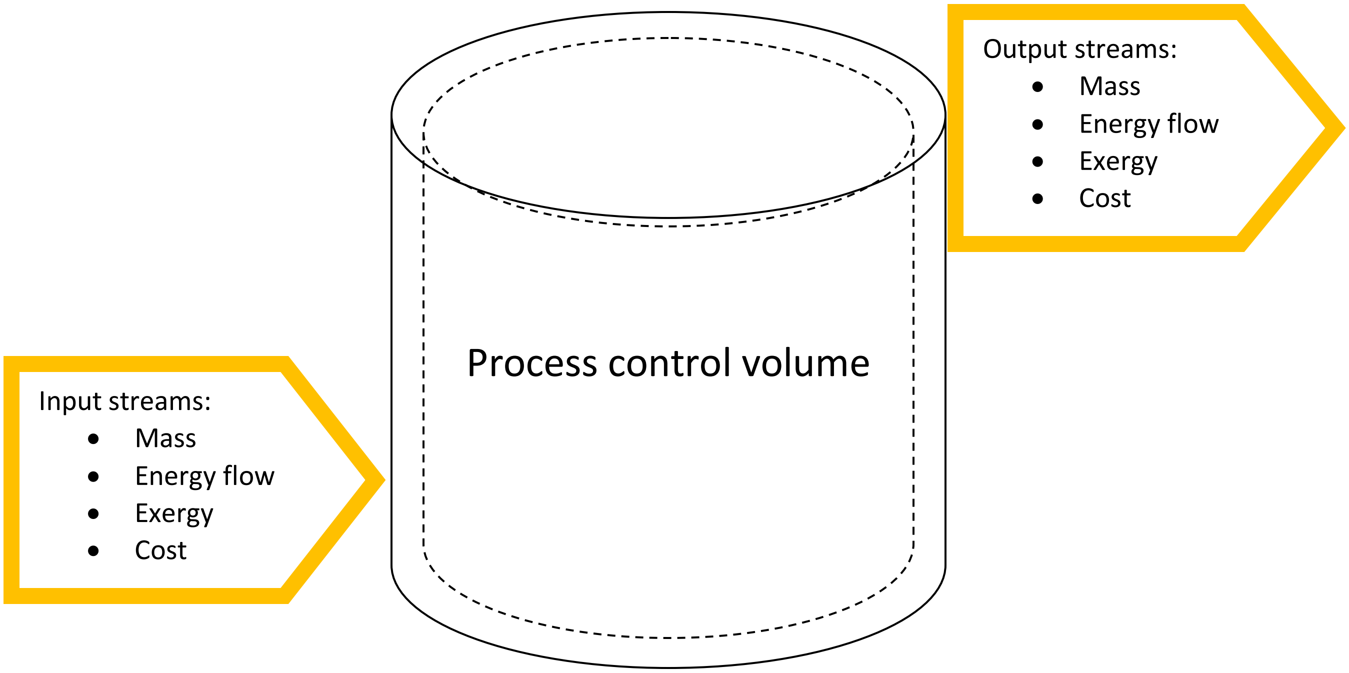


Figure (3) Assumptions of flow streams across any process.

3.2 Solar collectors

The collector is the heart of any solar energy system. The performance of such solar energy systems is largely dependent on the portion of solar insolation that is transferred to the fluid, which also depends on the working temperatures of this fluid. The useful portion of the insolation is a function of the design of the collector, its tilt and its orientation. It is also function of the design fluid temperatures and environmental parameters of the location such as air temperature, wind velocity, and insolation.

The solar collector instantaneous efficiency can be determined from its characteristic curve using the solar irradiance, mean collector and ambient temperatures. The curve used for FPC, evacuated tube, and CPC are expressed by Eq. (1) and the parameters given in Table 1. The corresponding efficiency equation for the medium-high temperatures parabolic trough collectors (PTC) is given by Eq. (2).

$$\eta_{col} = \eta_o - a_1\left(\frac{T_{co}-T_{amb}}{G_b}\right) - a_2\left(\frac{T_{co}-T_{amb}}{G_b}\right)^2 G_b \ldots.\ (1)$$

$$\eta_{col} = \eta_o - a_{11}(T_{co} - T_{amb}) - a_{21}\left(\frac{T_{co}-T_{amb}}{G_b}\right) - a_{31}\left(\frac{T_{co}-T_{amb}}{G_b}\right)^2 \ \ldots.\ (2)$$

Table 1: Efficiency parameters for different types of solar collectors.

Solar collector	a_1W/m^2	a_2 W/m^2	a_{11} W/m^2	a_{21} W/m^2	a_{31} W/m^2	η_o	Operating temp, °C
FPC	2.9	0.0108	--	--		0.768	80-100
CPC	0.59	0.0019	--	--		0.665	120-170
PTC	--	--	4.5×10^{-6}	0.039	3×10^{-4}	0.75	>170-300

The collector total area is estimated based on the collector energy balance equation as a function of collector efficiencies as;

$$A_{col} = \frac{Q_u}{\eta_{col}} G_b, m^2$$

Where Q_u is the collector useful thermal power and (G_b) is the global solar flux over the collector area, and A_{col} is the collector total area. The collector useful energy equation may exist according to the following relation;

$Q_u = \dot{m}_{col} \times \Delta H$ Where ΔH is the enthalpy difference across the collector in kJ/kg.

The exergy destruction across the solar collector is presented as following;

$$I_{collector} = A_{col} \times G_b \times \left(1 - \frac{T_{amb}}{T_{sun}}\right) + \dot{m}_{col}\left[h_i - h_o - T_{amb}(s_i - s_o)\right]_{col}$$

Where h represent the specific enthalpy, and s represents the specific entropy. Thermo-economic analysis for solar collector to any unit is presented as following;

$$C_{col-\ldots} = C_{q=0} + C_{\ldots-col} + Z_{col}^{IC\&OM}$$

And the product cost rate in $/kJ from solar collector field to any other unit (…);

$$c_{col-\ldots} = \frac{c_{\ldots-col}E_{\ldots-col} + Z_{col}^{IC\&OM}}{E_{col-\ldots}}$$

Where E is the exergy rate in kW.

3.3 Pump unit

Pump work W_p in kW is calculated as:

$$W_p = \frac{\dot{m}_{\dots} \times \Delta P}{\rho \times \eta_p}, kW$$

Where ΔP pressure difference between the condenser low pressure and the turbine high is pressure, and ρ is the density of the working fluid, and η_p is pump efficiency.
The pump outlet enthalpy is obtained via the following relation:

$$h_{out} = \frac{W_p}{\dot{m}} + h_{in}, kJ/kg$$

By knowing the environmental conditions (T_{amb}), the exergy destruction rate could be obtained from the following relation:

$$I_p = \dot{m}_p[h_{in} - h_{out} - T_{amb} \times (s_i - s_o)] + W_p, kW$$

And thermo-economically;

$$C_{p-\dots} = C_w + C_{\dots-p} + Z_p^{IC\&OM}$$

So; the unit product cost for the pump becomes;

$$c_{p-\dots} = \frac{c_w E_w + c_{\dots-p} E_{\dots-p} + Z_p^{IC\&OM}}{E_{p-\dots}}$$

Where, subscripts ($_{w,\, p}$) denotes for power in kW and pump.

3.4 Turbine unit

For models based on design approach, it is very important to find out the mass flow rate through the turbine by the knowing some parameters such as turbine efficiency, generator efficiency, and the developed power by the turbine. The outlet enthalpy of the turbine kJ/kg;

$$h_{out} = h_{in} - \eta_t \times \left(h_{in} - h_{out_s}\right), kJ/kg$$

Where η_t is the turbine efficiency and the subscript ($_{s,\, t}$) tends to isentropic state and turbine. The cycle flow rate kg/s is presented as following;

$$\dot{m}_t = \frac{W_t}{\eta_t \times \eta_{g\times}\left(h_{in} - h_{out_s}\right)}, kg/s$$

Where η_g is the generator efficiency.
Based on exergy analysis, the exergy destruction rate in kW is developed based on the following relation:

$$I_t = \dot{m}_t[h_{in} - h_{out} - T_{amb} \times (s_i - s_o)] - W_t, kW$$

Steam turbine would maintain one auxiliary equation for two streams outlet (power stream c_w, and exhaust stream to any followed unit). For this, the unit product power for steam turbine can be represented as follows;

$$c_w = \frac{C_{\dots-t}(E_{\dots-st} - E_{t-\dots}) + Z_t^{IC\&OM}}{E_w}, c_{\dots-st} = c_{t-\dots}$$

$$C_{t-\dots} + C_{w-\dots} = C_{\dots-t} + Z_t^{IC\&OM}$$

3.5 Heat exchanger units

Heat exchanger units include condensers, boiler heat exchangers, recuperators, and so on. For such units, it is recommended (based on design model) to assign the unit effectiveness (ϵ_{cond}). The following relations govern such units according to energy, exergy, and thermo-economic approaches.

Condenser heat rejection kW:

$$Q_{cond} = m\dot{}_{hot} \times \Delta h_{hot} = m\dot{}_{cold} \times \Delta h_{cold}, kW$$

$$\epsilon_{cond} = \frac{\Delta T_{cold}}{T_{hot_{max}} - T_{cold_{min}}}$$

Based on exergy analysis, the exergy destruction rate in kW is developed based on the following relation:

$$I_{cond} = m\dot{}_{hot}[h_{in} - h_{out} - T_{amb} \times (s_i - s_o)] + m\dot{}_{cold}[h_{in} - h_{out} - T_{amb} \times (s_i - s_o)], kW$$

Thermo-economically, recuperator example is demonstrated as following:

$$C_{rec-\cdots1} + C_{rec-\cdots2} = C_{\ldots1-rec} + C_{\ldots2-rec} + Z_{rec}^{IC\&OM}$$

$$c_{\ldots1-rec=}c_{rec-\cdots1}$$

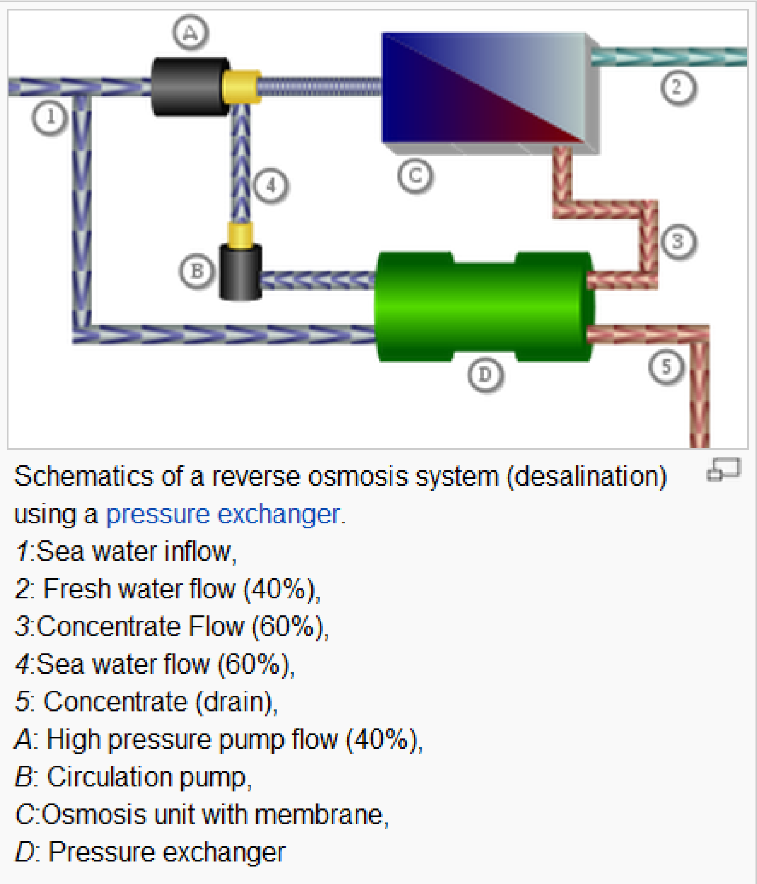
$$c_{\ldots2-rec=}c_{rec-\cdots1}$$

Where subscripts $_{\ldots1}$ denotes hot side stream from unit (1) and $_{\ldots2}$ denotes cold side stream from unit (2).

3.6 Reverse osmosis desalination unit

Reverse osmosis (RO) is a filtration method that removes many types of large molecules and ions from solutions by applying pressure to the solution when it is on one side of a selective membrane. The result is that the solute is retained on the pressurized side of the membrane and the pure solvent is allowed to pass to the other side. To be "selective," this membrane should not allow large molecules or ions through the pores (holes), but should allow smaller components of the solution (such as the solvent) to pass freely. Reverse osmosis is most commonly known for its use in drinking water purification from seawater, removing the salt and other substances from the water molecules. This is the reverse of the normal osmosis process, in which the solvent naturally moves from an area of low solute concentration, through a membrane, to an area of high solute concentration. The movement of a pure solvent to equalize solute concentrations on each side of a membrane generates a pressure and this is the "osmotic pressure." Applying an external pressure to reverse the natural flow of pure solvent, thus, is reverse osmosis. The process is similar to membrane filtration. However, there are key differences between reverse osmosis and filtration.

The predominant removal mechanism in membrane filtration is straining, or size exclusion, so the process can theoretically achieve perfect exclusion of particles regardless of operational parameters such as influent pressure and concentration. Reverse osmosis, however involves a diffusive mechanism so that separation efficiency is dependent on solute concentration, pressure and water flux rate. Formally, reverse osmosis is the process of forcing a solvent from a region of high solute concentration through a semipermeable membrane to a region of low solute concentration by applying a pressure in excess of the osmotic pressure. The membranes used for reverse osmosis have a dense barrier layer in the polymer matrix where most separation occurs. In most cases, the membrane is designed to allow only water to pass through this dense layer, while preventing the passage of solutes (such as salt ions). This process requires that a high pressure be exerted on the high concentration side of the membrane, usually 2–17 bar (30–250 psi) for fresh and brackish water, and 40–70 bar (600–1000 psi) for seawater, which has around 24 bar (350 psi) natural osmotic pressure that must be overcome. This process is best known for its use in desalination (removing the salt from sea water to get fresh water), but since the early 1970s it has also been used to purify fresh water for medical, industrial, and domestic applications. Figure (4) shows a schematic photograph of the RO process.



Schematics of a reverse osmosis system (desalination) using a pressure exchanger.
1:Sea water inflow,
2: Fresh water flow (40%),
3:Concentrate Flow (60%),
4:Sea water flow (60%),
5: Concentrate (drain),
A: High pressure pump flow (40%),
B: Circulation pump,
C:Osmosis unit with membrane,
D: Pressure exchanger

Figure (4) Schematic diagram of RO process.

The mathematical model for the proposed RO unit is written as follows:
The feed flow rate M_f based on recovery ratio *RR* and distillate flow rate M_d is:

$$M_f = \frac{M_d}{RR}$$

The distillate product salt concentration X_d;

$$X_d = X_f \times (1 - SR)$$

Where; X_f is the feed flow rate salt concentration, and SR is the salt rejection percentage; and the rejected brine is found from;

$$M_b = M_f - M_d$$

The rejected salt concentration kg/m^3 is estimated by;

$$X_b = \frac{M_f \times X_f - M_d \times X_d}{M_b}$$

The average salt concentration kg/m^{3}is estimated as;

$$X_{av} = \frac{M_f \times X_f + M_b \times X_b}{M_f + M_b}$$

The temperature correction factor TCF is found by the relation below;

$$TCF = exp\left(2700 \times \left(\frac{1}{273 + t} - \frac{1}{298}\right)\right)$$

The membrane water permeability k_w;

$$k_w = 6.84 \times 10^{-8} \times (18.6865 - (0.177 \times Xb))/(t + 273))$$

The salt permeability k_s is;

$$k_s = FF \times TCF \times 4.72 \times 10^{-7} \times \left(0.06201 - \left(5.31 \times 10^{-5} \times (t + 273)\right)\right)$$

Where FF is the membrane fouling factor. The calculations of osmotic pressure for feed side, brine side, and distillate product side are found as follows;

$$\Pi_f = 75.84 \times X_f$$
$$\Pi_b = 75.84 \times X_b$$
$$\Pi_d = 75.84 \times X_d$$

The average osmotic pressure on the feed side;

$$\Pi_{av} = 0.5 \times (\Pi_f + \Pi_b)$$

The net osmotic pressure across the membrane;

$$\Delta\Pi = \Pi_{av} - \Pi_d$$

The net pressure difference across the membrane;

$$\Delta P = \left(\frac{M_d}{3600 \times TCF \times FF \times A_e \times n_e \times n_v \times k_w}\right) + \Delta\Pi$$

Where A_e is the element area in m^2, ne is number of membrane elements, and n_v is the number of pressure vessels. The required high pressure pump power input in kW to the RO is estimated as;

$$HPP_{power} = \frac{1000 \times M_f \times \Delta P}{3600 \times \rho_f \times \eta_p}$$

Where ρ_f is the feed flow rate density, and η_p is the driving pump mechanical efficiency. The specific power consumption in kWh/m^3 is estimated as;

$$SPC = \frac{HPP_{power}}{M_d}$$

Based on exergy balance for RO section;

$$I_{RO} = W_{HPP} - \dot{m}_b \times \left(h_f - h_b\right) + \dot{m}_p \times \left(h_f - h_p\right), kW$$

Where h_f, h_b, and h_p is calculated based on seawater specific heat capacity, salinity X, and feed seawater temperature for each stream where;

$h_{f,p,b} = h_o + \left(A \times T + {}^{B}/_{2} \times T^2 + {}^{C}/_{3} \times T^3 + {}^{D}/_{4} \times T^4\right)$, Where; $h_o = 9.6296 \times X - 0.4312402 \times X^2$

And;

$$A = 4206.8 - 6.6197 \times X + 1.2288 \times 10^{-2} \times X^2$$
$$B = -1.1262 + 5.4178 \times 10^{-2} \times X - 2.2719 \times 10^{-4} \times X^2$$
$$C = 1.2026 - 5.3566 \times 10^{-4} \times X + 1.8906 \times 10^{-6} \times X^2$$
$$D = 6.8774 \times 10^{-7} + 1.517 \times 10^{-6} \times X - 4.4268 \times 10^{-9} \times X^2$$

Thermo-economic balance for the RO section;

$$C_p + C_{brine} = C_{feed} + Z_{RO}^{IC\&OM}$$

For Pelton wheel recovery turbine (PWT);

$$C_{w-HPP} + C_{brine-blowdown} = C_{brine-RO-PWT} + Z_{PWT}^{IC\&OM}$$

For pressure exchanger unit (PEX);

$$C_{feed-RO} + C_{brine-blowdown} = C_{feed} + C_{brine-RO-PEX} + Z_{PEX}^{IC\&OM}$$

By solving the above equations together, the following equation could maintain the overall thermo-economic balance of the system.

$$C_{brine-blowdown} + C_{product} = C_{cw-cond} + C_{q-col} + C_{w-pump} + Z_{total}^{IC\&OM}$$

3.7 Multi stage flash (MSF-BR) desalination unit

Multi-stage flash distillation (MSF) is a water desalination process that distills sea water by flashing a portion of the water into steam in multiple stages of what are essentially countercurrent heat exchangers. Multi-stage flash distillation plants produce over 85 percent of all desalinated water in the world, although a different type of desalinators, Reverse osmosis plants, are more numerous. The plant has a series of spaces called stages, each containing a heat exchanger and a condensate collector. The sequence has a cold end and a hot end while intermediate stages have intermediate temperatures. The stages have different pressures corresponding to the boiling points of water at the stage temperatures.

After the hot end there is a container called the brine heater. When the plant is operating in steady state, feed water at the cold inlet temperature flows, or is pumped, through the heat exchangers in the stages and warms up. When it reaches the brine heater it already has nearly the maximum temperature. In the heater, an amount of additional heat is added. After the heater, the water flows through valves back into the stages which have ever lower pressure and temperature. As it flows back through the stages the water is now called brine, to distinguish it from the inlet water. In each stage, as the brine enters, its temperature is above the boiling point at the pressure of the stage, and a small fraction of the brine water boils ("flashes") to steam thereby reducing the temperature until equilibrium is reached.

The resulting steam is a little hotter than the feed water in the heat exchanger. The steam cools and condenses against the heat exchanger tubes, thereby heating the feed water as described earlier. The total evaporation in all the stages is up to approx. 15% of the water flowing through the system, depending on the range of temperatures used. With increasing temperature there are growing difficulties of scale formation and corrosion. 120°C appears to be a maximum, although scale avoidance may require temperatures below 70°C. The feed water carries away the latent heat of the condensed steam, maintaining the low temperature of the stage. The pressure in the chamber remains constant as equal amounts of steam is formed when new warm brine enters the stage and steam is removed as it condenses on the tubes of the heat exchanger.

The equilibrium is stable, because if at some point more vapor forms, the pressure increases and that reduces evaporation and increases condensation. In the final stage the brine and the condensate has a temperature near the inlet temperature. Then the brine and condensate are pumped out from the low pressure in the stage to the ambient pressure. The brine and condensate still carry a small amount of heat that is lost from the system when they are discharged. The heat that was added in the heater makes up for this loss. The heat added in the brine heater usually comes in the form of hot steam from an industrial process co-located with the desalination plant. The steam is allowed to condense against tubes carrying the brine (similar to the stages). The energy that makes possible the evaporation is all present in the brine as it leaves the heater. The reason for letting the evaporation happen in multiple stages rather than a single stage at the lowest pressure and temperature, is that in a single stage, the feed water would only warm to an intermediate temperature between the inlet temperature and the heater, while much of the steam would not condense and the stage would not maintain the lowest pressure and temperature.

Such plants can operate at 23-27kWh/m3 (appr. 90MJ/m^3) of distilled water. Because the colder salt water entering the process counter flows with the saline waste water/distilled water, relatively little heat energy leaves in the outflow- most of the heat is picked up by the colder saline water flowing toward the heater and the energy is recycled. In addition, MSF distillation plants, especially large ones, are often paired with power plants in a cogeneration configuration. Waste heat from the power plant is used to heat the seawater, providing cooling for the power plant at the same time. This reduces the energy needed by one-half to two-thirds, which drastically alters the economics of the plant, since energy is by far the largest operating cost of MSF plants. Reverse osmosis, MSF distillation's main competitor, requires more pretreatment of the seawater and more maintenance, as well as energy in the form of work (electricity, mechanical power) as opposed to cheaper low-grade waste heat. Figure (5) shows a schematic diagram of MSF desalination process.

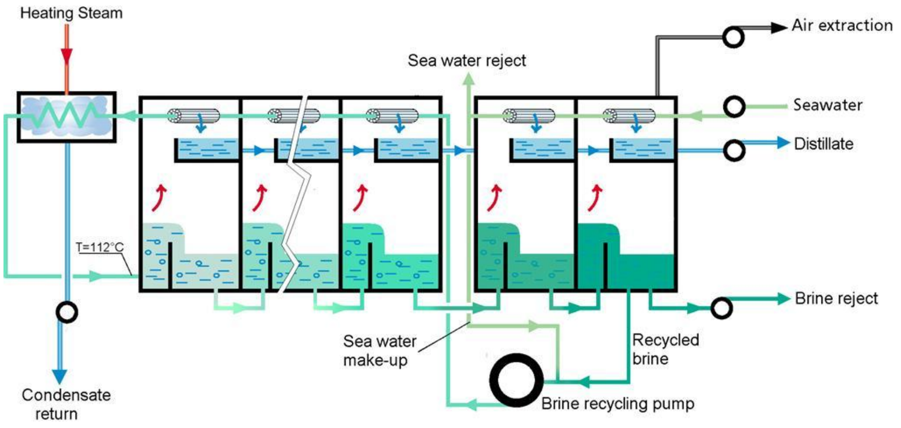


Figure (5) Schematic diagram of MSF brine recycle process.

The mathematical model based on design approach is presented as following:

For known distillate product, feed stream to the mixer unit is obtained:

$$M_f = (S_b/(S_b - S_f)) \times M_d$$

Total needed feed (M_{ft}) based on 1st splitter ratio:

$$M_{f_t} = M_f/SPL1$$

Therefore the rest of feed loss:

$$M_{fl} = M_{f_t} - \mathrm{M}_f$$

The brine blow-down loss:

$$M_b = M_f - M_d$$

Stage temp drop based on top brine temperature (TBT), last stage brine temperature (T_n) and number of stages (N):

$$T_{stg} = TBT - T_n/N$$

The recycle brine flow rate M_r and latent heat L is then calculated:

$$Y = (C_p \times T_{stg})/L$$

$$M_r = M_d/(1-(1-Y)^N)$$

Then the salinity of the recycle stream is calculated S_r:

$$S_r = \frac{(S_f \times M_f + (M_r - M_d) \times S_b - M_b \times S_b)}{M_r}$$

The outlet temperature of the distillate product T_d could be calculated based on brine blow down temperature T_n, non equilibrium allowance *NEA*, and boiling point ratio *BPR*.

$$T_d = T_n - NEA - BPR$$

The non-equilibrium allowance *NEA* and *BPR* are calculated by the following equations;

$$NEA = A + B \times T_n + C \times T_n^2 + D \times T_n^3$$

Where A=2.556, B=-0.203×10^{-1},$C = -0.129 \times 10^{-1}$ and $D = 0.1123 \times 10^{-5}$

$$BPR = (B + C \times S) \times S$$

Where S is the stream salinity and,

$$10^3 \times B = 6.71 + 6.43 \times 10^{-2} \times T_n + 9.74 \times 10^{-5} \times T_n^2$$

$$10^5 \times C = 2.38 + 9.59 \times 10^{-3} \times T_n + 9.42 \times 10^{-5} \times T_n^2$$

For the heat recovery and rejection sections, the overall heat transfer coefficient based on vapor temperature T_v:

$$U = 1.7194 + 3.2063E - 3 \times T_v + 1.5971E - 5 \times T_v^2 - 1.9918E - 7 \times T_v^3$$

The exergy destruction balance across the MSF plant can be introduced as following;

$$I = W_p + E_{bo} + E_{fi} - E_{fo} - E_d, kW$$

Where, W_p is the total pumping power required, E is the exergy rate, and subscripts ($_{bo,\ fi,\ fo}$) denotes to TBT, inlet feed, and outlet feed streams respectively. Thermo-economically, the balanced equations should be presented as follows;

$$C_d + C_{brine} + C_{condi} = C_{condo} + C_{fi} + Z_{msf}^{IC\&OM}$$

Where C_d is the distillate product cost \$/h, C_{brine} is the brine blow down cost and is specified as zero cost, and C_{fi}. The unit specific cost of inlet seawater feed stream to the MSF condenser is considered the same as outlet preheated stream entered the condenser/brine-heater unit ($c_{cwi\text{-}cond}$=c_{fi}). So the relation would become as follows;

$$c_d = \frac{c_{condo}E_{condo} - c_{fi}\Delta E_f + Z_{msf}^{IC\&OM}}{E_d}$$

3.8 Multi effect distillation desalination unit

Multiple-effect distillation (MED) is a distillation process often used for sea water desalination. It consists of multiple stages or "effects". In each stage the feed water is heated by steam in tubes. Some of the water evaporates, and this steam flows into the tubes of the next stage, heating and evaporating more water. Each stage essentially reuses the energy from the previous stage. The tubes can be submerged in the feed water, but more typically the feed water is sprayed on the top of a bank of horizontal tubes, and then drips from tube to tube until it is collected at the bottom of the stage. The plant can be seen as a sequence of closed spaces separated by tube walls, with a heat source in one end and a heat sink in the other end. Each space consists of two communicating subspaces, the exterior of the tubes of stage n and the interior of the tubes in stage n+1. Each space has a lower temperature and pressure than the previous space, and the tube walls have intermediate temperatures between the temperatures of the fluids on each side. The pressure in a space cannot be in equilibrium with the temperatures of the walls of both subspaces. It has an intermediate pressure. Then the pressure is too low or the temperature too high in the first subspace and the water evaporates. In the second subspace, the pressure is too high or the temperature too low and the vapor condense. This carries evaporation energy from the warmer first subspace to the colder second subspace. At the second subspace the energy flows by conduction through the tube walls to the colder next space. Figure (6) shows a schematic diagram of the MED process. There are some features of such kind of desalination process.

- Low energy consumption (less than 1.0 kWh/m^3) compared to other thermal processes.
- Operates at low temperature (< 70 °C) and at low concentration (< 1.5) to avoid corrosion and scaling.
- Does not need pre-treatment of sea water and tolerates variations in sea water conditions.
- Highly reliable and simple to operate.
- Low maintenance cost.
- 24 hour a day continuous operation with minimum supervision.
- Can be adapted to any heat source including hot water or waste heat from power generation or industrial processes.

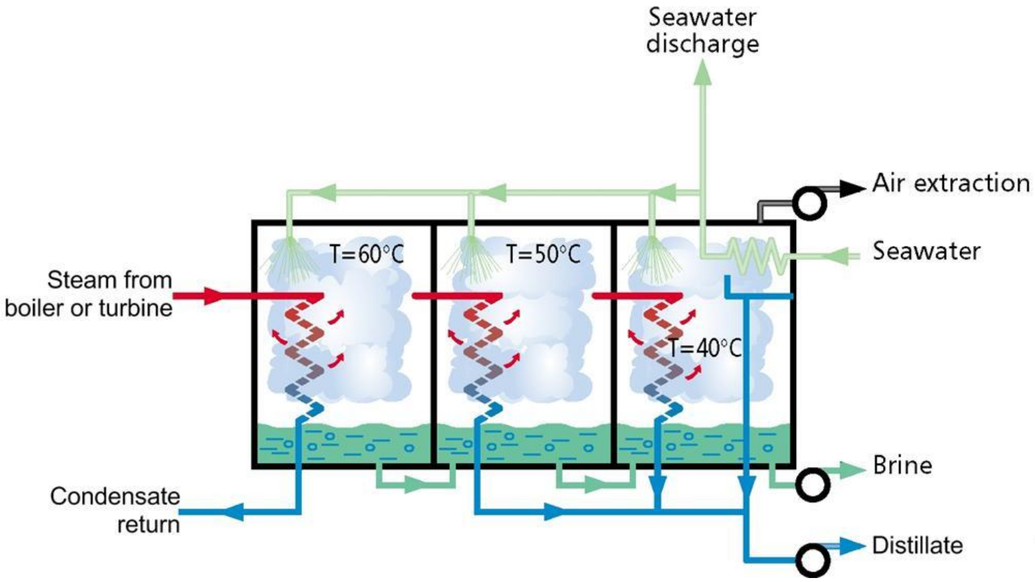


Figure (6) Schematic diagram of multi effect distillation process.

The analysis introduced in MED section, is presented based on single effect evaporation module. The first method used to desalt seawater in large quantities was the single effect desalting system consisting of an evaporator-condenser combination (Fig. 7). Single effect evaporation (SEE) has limited industrial applications. The system is used in marine vessels and this because the system has thermal performance ratio less than one, i.e.; the amount of water produced is less than the amount of heating steam used to operate the system. The main components of the unit are the evaporator and the feed pre-heater condenser. The evaporator consist of an evaporator\condenser heat exchanger tubes, a vapor space, un-evacuated water pool, a line for removal of non condensable gases, a water distribution system, and a mist eliminator. A heat source (steam *S*) heats the incoming feed *F* to the evaporator from its entering feed temperature T_f to its boiling temperature T_b, and

evaporates part of it equal to *D*. The vapor *D* is directed to the condenser where it condenses and heats the cooling water M_{sea} from seawater temperature T_{sea} to the feed temperature T_f. Part of M_{sea} leaving the condenser is used as feed *F* while the balance *B* (=M_{sea}-*F*) is called brine blow-down and is rejected back to the sea. In the condenser of a single-effect desalting system, a small portion of the latent heat given off by condensing the vapor *D* is utilized to heat the feed-water *F*, while the balance $D\times L=F\times C_p\times (T_f\text{-}T_{sea})$ is rejected back to sea. The mathematical model for this type is illustrated as following:
Energy balance for the condenser unit based on the specified effectiveness ε:

$$T_f=\varepsilon\times(T_v\text{-}T_{sea})+T_{sea}$$

where T_v is the vapor temperature;
The distillate temperature is obtained from the same equation:

$$T_d=T_v\text{-}(\ \varepsilon\times\ (T_v\text{-}T_{sea})$$

Mass and material balances;

$$M_f=M_d\times\ S_b/(S_b\text{-}S_f)$$
$$M_b=M_d\times\ S_f/(S_b\text{-}S_f)$$

And steam flow rate M_s could be obtained from the following relation;
$M_s=M_d/PR$ where *PR* is the performance ratio which is also obtained as following;

$$PR=\frac{L(T_s)}{L(T_v)+C_p(T_{av},S_f)\times(T_v-T_f)\times\dfrac{S_b}{(S_b-S_f)}+\dfrac{S_f}{S_b-S_f}\times C_p(T_{av},S_f)BPE(T_b,S_b)}$$

Where, *BPE* is the boiling point elevation as a function of brine temperature and salinity percent;
Cooling water blow down from the condenser unit is obtained from the following energy balance relation;

$$M_{cw}=\frac{M_d\times L(T_v)}{C_p(T_{av},S_f)\times(T_f-T_{sea})}-M_f$$

Where the T_{av} is the average temperature for the feed seawater across the condenser unit ($T_{av}=\frac{T_f-T_{sea}}{2}$)
Therefore; the total mass flow rate is then calculated;

$$M_{ft}=M_{cw}+M_f$$

Heat Transfer areas for condenser and evaporator units (A_c, A_e) are obtained based on logarithmic mean temperature *LMT*, latent heat *L* and overall heat transfer coefficient *U;*
For condenser unit;

$$LMT_c=\frac{(T_f-T_{sea})}{log\dfrac{(T_v-T_{sea})}{(T_v-T_f)}}$$

$$A_c=\frac{M_d\times L(T_v)}{U_c(T_v)\times LMT_c}$$

$$A_e=\frac{M_d\times\dfrac{S_b}{(S_b-S_f)}\times C_p(T_b,S_b)\times(T_b-T_f)+M_d\times L(T_v)}{U_e(T_s-T_b)}$$

Thermo-physical properties and overall heat transfer coefficient are calculated form the following correlations:

- Boiling point elevation:

$$\begin{aligned}BPE&=0.0825431+0.0001883\times T+0.00000402\times T^2\times S-0.0007625+0.0000902\times T\\&\quad-0.00000052\times T^2\times S^2+0.0001522-0.000003\times T-0.00000003\times T^2\times S^3\end{aligned}$$

- Specific heat capacity:

$a = 4206.8 - 6.6197 \times S + 1.2288 \times 10^{-2} \times S^2$
$b = -1.1262 + 5.4178 \times 10^{-2} \times S - 2.2719 \times 10^{-4} \times S^2$
$c = 1.2026 \times 10^{-2} - 5.3566 \times 10^{-4} \times S + 1.8906 \times 10^{-6} \times S^2$
$d = 6.8774 \times 10^{-7} + 1.517 \times 10^{-6} \times S - 4.4268 \times 10^{-9} \times S^2$
$Cp = a + b \times T + c \times T^2 + d \times T^3$

- Latent heat of vaporization:

$L = 2501.897149 - 2.407064037 \times T + 1.192217 \times 10^{-3} \times T^2 - 1.5863 \times 10^{-5} \times T^3$

- The overall heat transfer coefficients for evaporator and end condenser units:

$Ue = 1.9695 + 1.2057 \times 10^{-2} \times T - 8.5989 \times 10^{-5} \times T^2 + 2.5651 \times 10^{-7} \times T^3$
$Uc = 1.7194 + 3.2063 \times 10^{-3} \times T + 1.5971 \times 10^{-5} \times T^2 - 1.9918 \times 10^{-7} \times T^3$

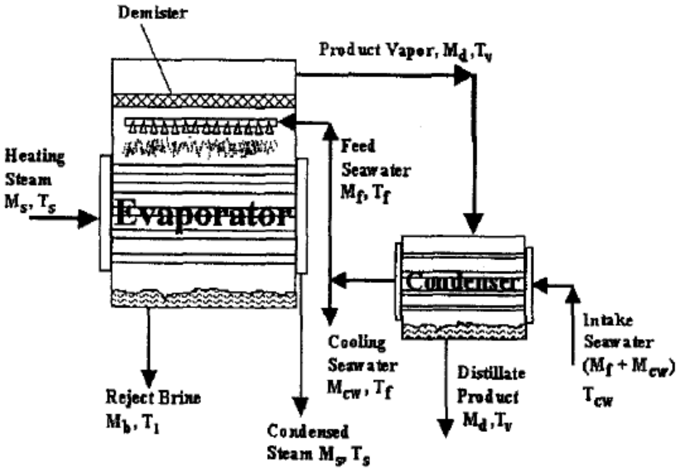

Demister
Product Vapor, M_d, T_v
Heating Steam M_s, T_s
Evaporator
Feed Seawater M_f, T_f
Condenser
Intake Seawater $(M_f + M_{cw})$ T_{cw}
Cooling Seawater M_{cw}, T_f
Distillate Product M_d, T_v
Reject Brine M_b, T_1
Condensed Steam M_s, T_s

Figure (7) Schematic display of Single Effect Evaporation (SEE) process.

The exergy destruction rate is calculated based on the overall exergy balance equation:

$$I = W_{fp} + W_{bp} + W_{dp} + E_{si} - E_{so} + E_f + E_d, kW$$

Where, W and E represent the pumping power and exergy streams. And subscripts ($_{fp,\ bp,\ dp,\ s}$) denotes the feed pump, brine pump, distillate pump and steam respectively. Thermo-economically, the MED process streams can be presented as follows;

$$C_d + C_{brine} + C_{steam-p} = C_{steam-med} + C_{fi} + Z_{med}^{IC\&OM}$$

Where C_d is the distillate product cost $/h, C_{brine} is the brine blow down cost and is specified as zero cost, and C_{fi}. So the relation for thermo-economic distillate cost would become as follows;

$$c_d = \frac{c_{fi}\,Ex_{fi} + c_{steam-med}\,\Delta Ex_{steam} + Z_{med}^{IC\&OM}}{Ex_d}$$

3.9 Multi effect distillation thermal vapor compression desalination unit

The MED-TVC evaporator is basically an MED evaporator fitted with a *thermo-compressor*. The purpose of the thermo-compression of the vapor is to take advantage of the pressure of the available steam, when this pressure is sufficient (i.e. above 2 bar abs), to enhance the units' performance. The incoming steam, called motive steam, is fed into the thermo-compressor through a sonic nozzle. Its expansion will allow low pressure steam from a cell of the evaporator to be sucked out. Both steams will be mixed in the thermo-compressor body. The mixture is then compressed to the pressure of the first bundle through a shock wave. The latent heat of the sucked vapor is thus recycled in the evaporator and is again available for desalination, leading to energy savings. The performance of a thermo-compressor is expressed by the mass of sucked steam (in kg) per kg of motive steam. Figure (8) shows a schematic diagram of multi effect distillation-thermal vapor compression process.

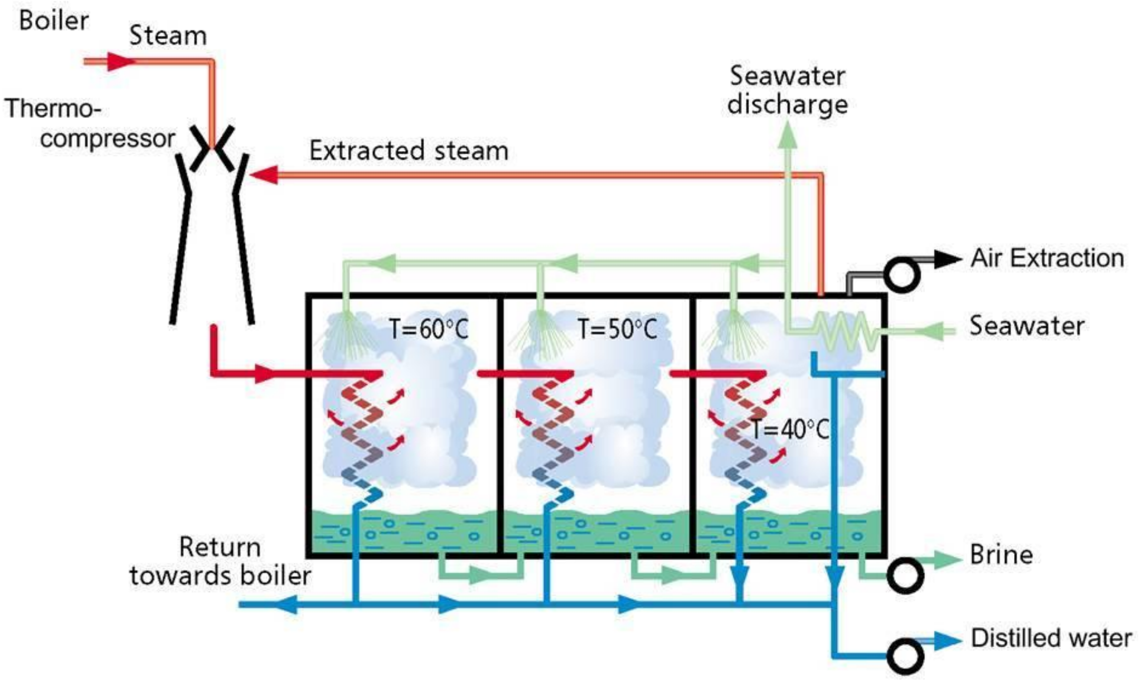


Figure (8) A schematic diagram of multi effect distillation thermal vapor compression process.

The most important and critical step in modeling the TVC desalination system is the evaluation of the performance of the steam jet ejector. The main data required from analyzing the steam jet ejector is the determination of the mass of motive steam required per unit mass of the entrained vapor (Ra), given the pressure of the motive steam (P_{ms}), discharge pressure (P_d) and the suction pressure or entrained vapor pressure (P_{ev}). There are a limited number of methods available in the literature to analysis the steam jet ejector. However, these methods require tedious and lengthy calculation procedures. Additionally, most of these methods are based on using many correction factors that depend heavily on the detail design of the ejector. El-

Dessouky, 1997, developed the following relationships to evaluate the performance of the steam jet ejector. The mathematical model of MED-TVC parallel feed configuration with feed water heaters is introduced as follows:
Last effect vapor temperature is function of blow down brine temperature:

$$T_v = T_b - BPE$$

And the temperature drop between effects is then obtained based on the number of effects:

$$\Delta T = (T_s - T_v)/N_{eff}$$

Where, T_s is the steam temperature, T_v is the vapor temperature.
Brine flow temperature profile is then obtained:

$$T_{bn} = T_b + (N_{eff} - 1)\Delta T$$

The corresponding saturation vapor pressure P_v kPa is obtained as follows:

$$P_v = -1.335e - 17 \times T_v^9 + 1.328e - 14 \times T_v^8 - 5.605e - 12 \times T_v^7 + 1.3e - 9 \times T_v^6 - 1.786e - 7 \times T_v^5 + 1.585e - 5 \times T_v^4 - 0.0007782 \times T_v^3 + 0.02466 \times T_v^2 - 0.2679 \times T_v + 1.739$$

The pressure drop in the demister in kPa/m (based on backing density ρ, vapor velocity V, wire diameter d, length of packing L_p) is evaluated from the correlation below:

$$P_p = 3.88178 \times (\rho_p^{0.375798}) \times (V_v^{0.81317}) \times (d_w^{-1.56114147}) \times (L_p/1000)$$

The vapor pressure past the demister in kPa is then calculated:

$$P_{ev} = P_v - P_p$$

For mass balance and material the distillate profile is calculated according to 1st effect productivity as a function of latent heat and number of effects:

$$D_1 = \frac{D_t}{L(T_{v1}) \times \sum 1/L(T_{v_{1:n}})}$$

Then the distillate profile can be calculated:

$$D_{1:n} = \frac{D_1 \times L(T_{v1})}{L(T_{v_{1:n}})}$$

Brine and salt balances could be calculated:

$$M_b = D_t(S_f/(Sb - Sf)$$

The motive steam mass flow rate that entered the steam ejector is then calculated from the energy balance across the 1st effect:

$$M_{ms} = \frac{(M_f \times C_p(T_f, S_f) \times (T_b - T_f) + D_1 \times L(T_{v1})}{L(T_s) \times (1 + \frac{1}{Ra})}$$

The amount of entrained vapor mass flow rate M_{ev} is then calculated from the entrainment ratio (Ra):

$$M_{ev} = M_{ms}/Ra$$

The total steam mass flow rate to evaporator:

$$M_{st} = M_{ev} + M_{ms}$$

The system performance parameters are calculated as follows:

$$PR = D_t/M_{ms}$$

Heat Transfer areas H.T.A for the evaporator is calculated based on the evaporator thermal load Q_e:

$$Q_e = (M_{ms} + M_{ev}) \times L(T_s)$$

$$A_e = \frac{Q_e}{U_e(T_b) \times (\Delta T - BPE(T_b))}$$

Where, BPE is the boiling point elevation as a function of last effect brine temperature (T_b).

Heat transfer area for feed heaters A_{fh} is calculated based on LMT:

$$A_{fh} = \frac{M_f \times C_p(\Delta t, S_f) \times \Delta t}{U_e(\Delta t) \times LMT_{fh}}$$

The steam ejector mathematical model is obtained based on the following model:

The pressure of compressed vapor kPa P_s is function of vapor pressure and compression ratio (Cr) and the entrained vapor from effect number n:

$$P_s = Cr \times P_{ev}$$

The high velocity at Nozzle exit can be calculated based on enthalpy difference $(h_{si} - h_{so})$ across the ejector nozzle:

$$V_{ne} = \sqrt{(2000 \times (h_{si} - h_{so})}$$

The entrainment ratio is obtained as following by calculating temperature and pressure correction factors:

$$PCF = 3 \times 10^{-7} \times P_{ms}^2 - 0.0009 \times P_{ms} + 1.6101$$

Where, P_{ms} is the motive steam pressure in kPa, and T_v is the vapor temperature from effect number n.

$$TCF = 2 \times 10^{-8} \times T_v^2 - 0.0006 \times T_v + 1.0047$$

$$Ra = 0.296 \times \frac{P_s^{1.19}}{P_{ev}^{1.04}} \times \left(\frac{P_{ms}}{P_{ev}}\right)^{0.015} \times \left(\frac{PCF}{TCF}\right)$$

The expansion ratio:

$$Er = \frac{P_{ms}}{P_{ev}}$$

The nozzle cross section area (A_1) based on the nozzle diameter D_n:

$$A_1 = \frac{\pi D_n^2}{4}$$

The area ratios (A_1/ A_3 & A_2/ A_1) and the areas (A_3 & A_2) of the nozzle outlet and the diffuser are then calculated:

$$\frac{A_1}{A_3} = 0.34 \times P_s^{1.19} \times P_{ms}^{-1.12} \times Ra^{-0.16}$$

$$\frac{A_2}{A_1} = 1.04 \times P_s^{-0.83} \times P_{ms}^{0.86} \times Ra^{-0.12}$$

Nozzle outlet area m^2:

$$A_3 = \frac{A_1}{\frac{A_1}{A_3}}$$

Diffuser area m^2:

$$A_2 = A_1 \times \frac{A_2}{A_1}$$

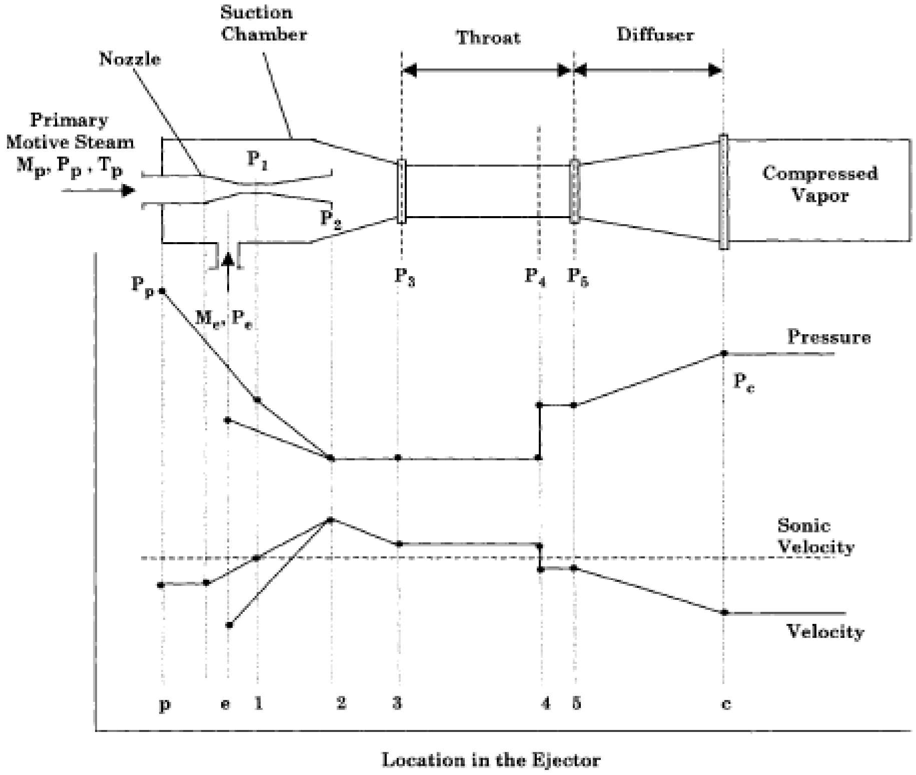


Figure (10) Pressure and velocity profiles inside the steam jet ejector.

3.10 Multi effect distillation mechanical vapor compression desalination unit

Mechanical vapor-compression desalination process (MVC) is the most attractive and valuable among different single stage desalination processes. The MVC system is compact, confined, and does not require external heating source, which is opposite to thermal, absorption, or adsorption vapor compression. The system is driven by electric power; therefore, it is suitable for remote population areas with access to power grid lines. Another advantage of the MVC system is the absence of the down condenser and the cooling water requirements. This is because the compressor operates on the entire vapor formed within the system. Other advantages of the system include:

- Moderate investment cost.
- Proven industrial reliability to long lifetime operation.
- Simple seawater intake and pretreatment.
- The system adopts the horizontal falling film tube configuration, which allows for high heat transfer coefficient.
- The low temperature operation, 60°C, allows for reduced scaling and heat losses and minimum requirement of thermal insulation.
- The system is modular type and it is simple to enlarge production volume by adopting additional modules.
- High product purity.

- Simple system adjustment to load variations, through temperature manipulation.
- Provide low power to water ratio without reducing the fuel utilization potential.

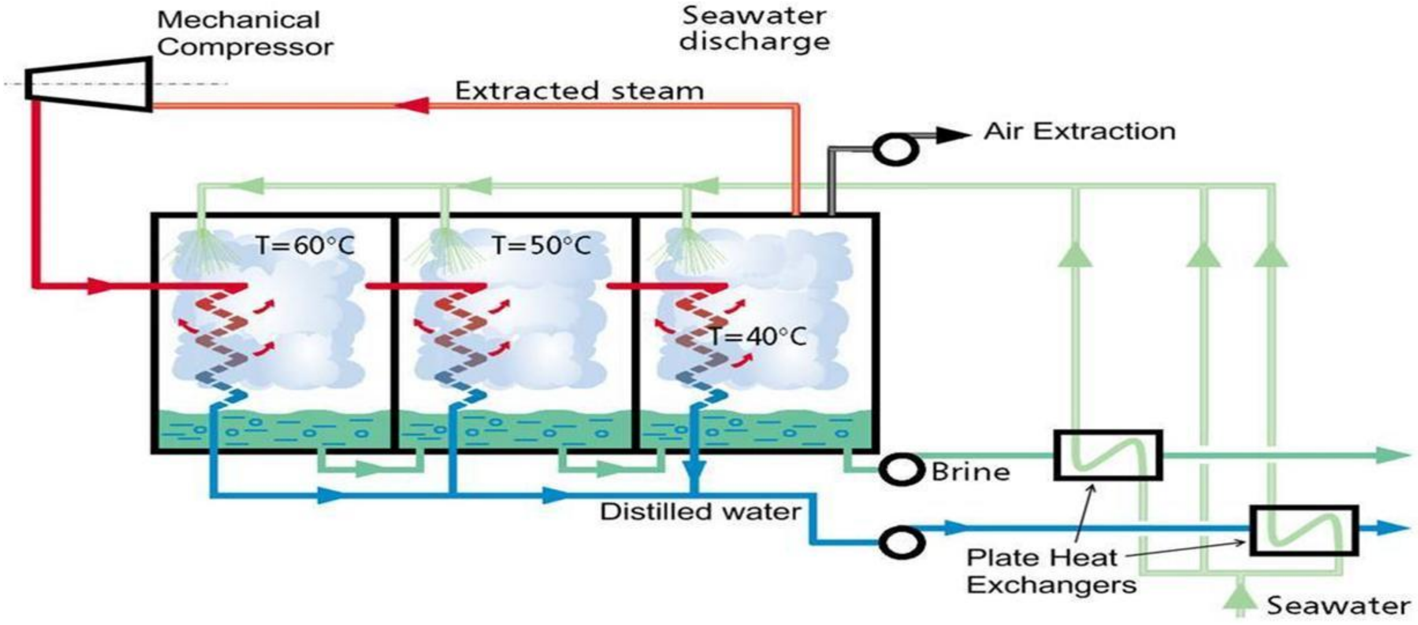


Figure (9) A schematic diagram of multi effect distillation mechanical vapor compression process.

The mathematical model of MED part considered the same as that presented in MED-TVC. Therefore, the model is pinpointed on the mechanical vapor compressor unit. The specific volume of inlet vapor at brine temperature is obtained as follows:

$$v_{comp_i} = 163.3453 - 8.04142 \times T + 0.17102 \times T^2 - 1.87812 \times 10^{-3} \times T^3 + 1.03842 \times 10^{-5} \times T^4 - 2.28215 \times 10^{-8} \times T^5$$

The compression ratio:

$$Cr = \frac{P_{comp_o}}{P_{comp_i}}$$

The compressor horse power needed:

$$\dot{W}_{comp} = M_s \times \frac{\gamma}{\eta_{comp} \times (\gamma - 1)} \times (P_{comp_i} \times v_{comp_i}) \times \left(\frac{P_{comp_o}}{P_{comp_i}}\right)^{\left(\frac{\gamma-1}{\gamma}\right)}$$

Where, M_s is the steam mass flow rate in kg/s, γ is the isentropic index, and P_{comp} is the pressure in kPa.

4. MatLab/Simulink

4.1 Introduction

Simulink® software models, simulates, and analyzes dynamic systems. It enables user to pose a question about a system, model the system, and see what happens. With Simulink, user can easily build models from scratch, or modify existing models to meet your needs. Simulink supports linear and nonlinear systems, modeled in continuous time, sampled time, or a hybrid of the two. Systems can also be multi rate—having different parts that are sampled or updated at different rates. Thousands of scientists and engineers around the world use Simulink to model and solve real problems in a variety of industries, including:

- Aerospace and Defense
- Automotive
- Communications
- Electronics and Signal Processing
- Medical Instrumentation

With Simulink, user can move beyond idealized linear models to explore more realistic nonlinear models, factoring in friction, air resistance, gear slippage, hard stops, and the other things that describe real-world phenomena. Simulink turns your computer into a laboratory for modeling and analyzing systems that would not be possible or practical otherwise.

Whether you are interested in the behavior of an automotive clutch system, the flutter of an airplane wing, or the effect of the monetary supply on the economy, Simulink provides you with the tools to model and simulate almost any real-world problem. Simulink also provides demos that model a wide variety of real-world phenomena.

Simulink provides a graphical user interface (GUI) for building models as block diagrams, allowing user to draw models as you would with pencil and paper. Simulink also includes a comprehensive block library of sinks, sources, linear and nonlinear components, and connectors. If these blocks do not meet your needs, however, you can also create your own blocks. The interactive graphical environment simplifies the modeling process, eliminating the need to formulate differential and difference equations in a language or program.

Models are hierarchical, so you can build models using both top-down and bottom-up approaches. User can view the system at a high level, and then double-click blocks to see increasing levels of model detail. This approach provides insight into how a model is organized and how its parts interact. Model analysis tools include linearization and trimming tools, which can be accessed from the MATLAB command line, plus the many tools in MATLAB and its application toolboxes. Because MATLAB and Simulink are integrated, you can simulate, analyze, and revise your models in either environment at any point.

Simulink software is tightly integrated with the MATLAB environment. It requires MATLAB to run, depending on it to define and evaluate model and block parameters. Simulink can also utilize many MATLAB features. For example, Simulink can use the MATLAB environment to:

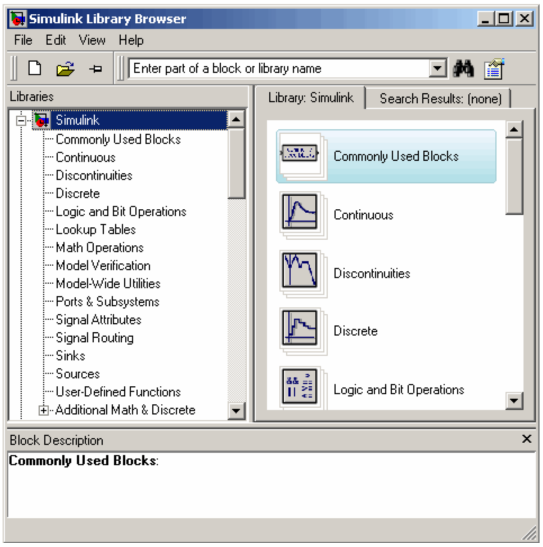
- Define model inputs.
- Store model outputs for analysis and visualization.
- Perform functions within a model, through integrated calls to MATLAB operators and functions.

4.2 Simulink software basics: starting & interface

User MATLAB environment must be running before you can open Simulink software. User starts Simulink from within MATLAB.

To start Simulink and open the Simulink Library Browser:

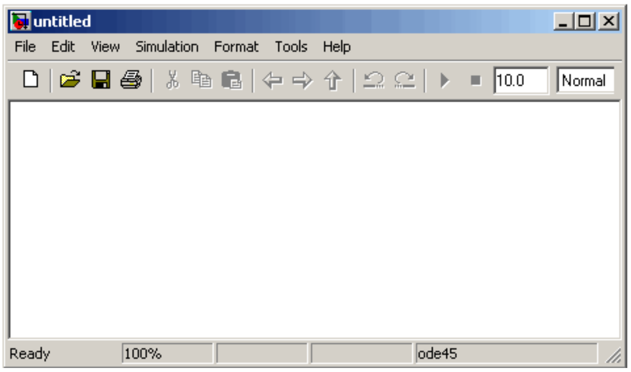
1. Start MATLAB. For more information, see Starting a MATLAB Session in the MATLAB Getting Started Guide.
2. Enter simulink in the MATLAB Command Window.
   The Simulink Library Browser opens.



User can open existing Simulink models or create new models from the Simulink Library Browser.

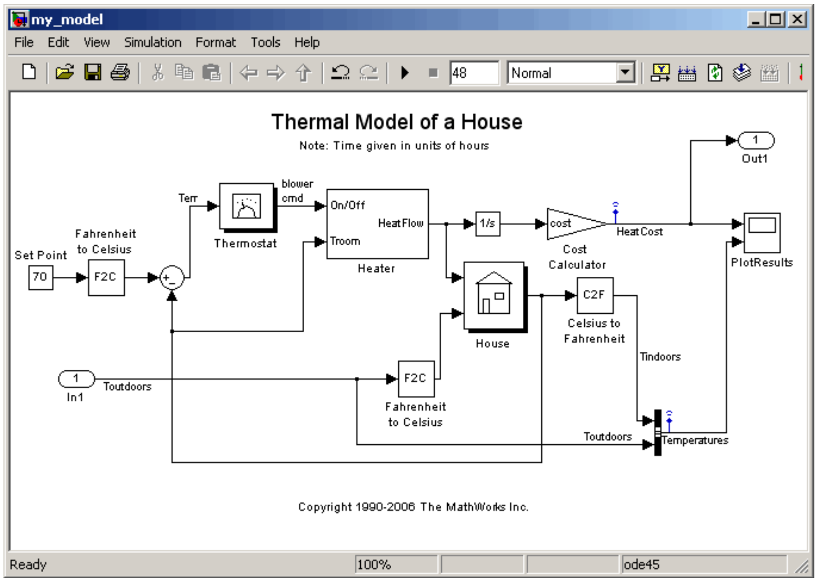
To create a new model:

- Select File > New > Model in the Simulink Library Browser.
  The software opens an empty model window.

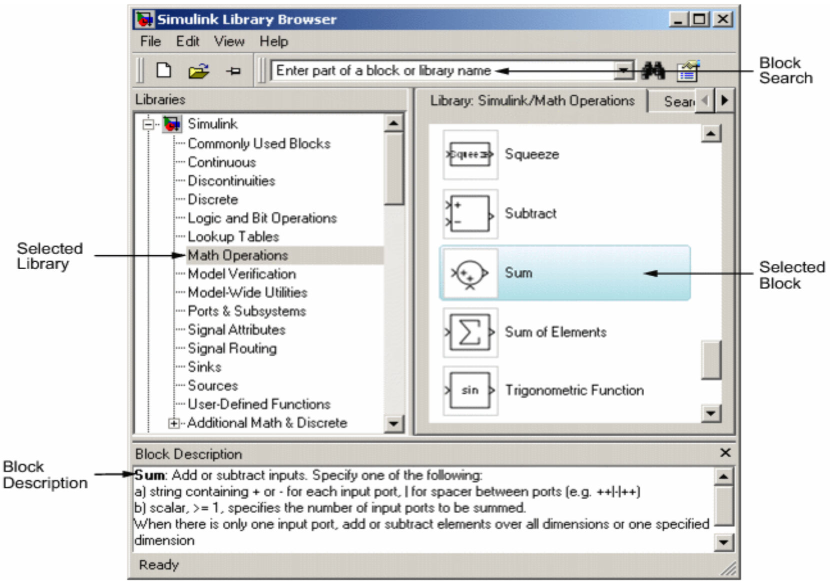


To open an existing model:

1. Select File > Open in the Simulink Library Browser.
   The Open dialog box appears.
2. Select the model (.mdl file) you want to open, and then click Open.
   The software opens the selected model in the model window.



The Library Browser displays the Simulink block libraries installed on user system. User builds models by copying blocks from a library into a model window.



Sum: Add or subtract inputs. Specify one of the following:
a) string containing + or - for each input port, | for spacer between ports (e.g. ++|-|++)
b) scalar, >= 1, specifies the number of input ports to be summed.
When there is only one input port, add or subtract elements over all dimensions or one specified dimension

When using the Library Browser, note the following:

- User can view the blocks in a library by selecting the library name on the left side of the Library Browser, or by double-clicking the library.
- When user selects a block, a description of that block appears at the bottom of the browser.
- For more information on a block, select the block, then select Help > Help on the Selected Block to display the help page for the block.
- User can view the parameters for a block by right-clicking the block, then selecting Block Parameters.

4.3 Masking systems

Masking the model is very important to make it easy to handle and attractive. A mask is a custom user interface for a subsystem that hides the subsystem's contents, making it appear to the user as an atomic block with its own icon and parameter dialog box. Note that this is different from an Atomic Subsystem, which the Simulink software treats as a unit when determining the execution order of block methods. Masking a subsystem provides only graphical, not functional, grouping. The Simulink Mask Editor enables you to create a mask for any subsystem. Masking a subsystem allows you to:

- Replace the parameter dialogs of a subsystem and its contents with a single parameter dialog with its own block description, parameter prompts, and help text.
- Replace a subsystem's standard icon with a custom icon that depicts its purpose.
- Prevent unintended modification of subsystems by hiding their contents behind a mask.
- Create a custom block by encapsulating a block diagram that defines the block's behavior in a masked subsystem and then placing the masked subsystem in a library.

Masks can include any of the following features:

Mask Icon

The mask icon replaces a subsystem's standard icon, i.e., it appears in a block diagram in place of the standard icon for a subsystem block. The Simulink product uses MATLAB code that you supply to draw the custom icon. You can use any of the MATLAB drawing commands in the icon code. This gives you great flexibility in designing an icon for a masked subsystem.

Mask Parameters

You can define a set of user-settable parameters for a masked subsystem. The value of a parameter is stored in the mask workspace (see Mask Workspace) as the value of a variable whose name you specify. These associated variables allow you to link mask parameters to specific parameters of blocks inside a masked subsystem (internal parameters) such that setting a mask parameter sets the associated block parameter (see Linking Mask Parameters to Block Parameters).

Mask Parameter Dialog Box

The mask parameter dialog box contains controls that enable a user to set the values of the mask's parameters and hence the values of any internal parameters linked to the mask parameters. The mask parameter dialog box replaces the subsystem's standard parameter dialog box, i.e., clicking on the masked subsystem's icon causes the mask dialog box to appear instead of the standard parameter dialog box for a Subsystem block. User can customize every feature of the mask dialog box, including which parameters appear on the dialog box, the order in which they appear, parameter prompts, the controls used to edit the parameters, and the parameter callbacks (code used to process parameter values entered by the user).

Mask Initialization Code

The initialization code is MATLAB code that you specify and that the Simulink software runs to initialize the masked subsystem at critical times, such as updating the model or starting simulation. You can use the initialization code to set the initial values of the masked subsystem's mask parameters. See Initialization Pane for more information.

4.4 Building solar desalination model by simulink: **MED-PF-TVC** Case study

First of all, user should build his own model on a scratched paper. Then it is very easy to convert the idea related to the typed process to become a useful simulink model. For example, suppose that user wants to build solar desalination plant containing the following:

- ✓ Solar radiation model to calculate the solar intensity in W/m^2.
- ✓ Solar collector (FPC, CPC, PTC).
- ✓ Boiler heat exchanger unit (BHX).
- ✓ Pumps.
- ✓ MED-PF-TVC configuration.

Also, user should specify the following:

- ✓ Types of working fluid (Toluene, Water, seawater, Therminol-VP1, Butane …).
- ✓ Environmental operating conditions (solar radiation, ambient temperature, wind velocity …).
- ✓ Location specifications (latitude, longitude, solar angles…).
- ✓ Desalination plant specifications (salinity, productivity…).

The second part is to model the process related to user scratch via simulink browser and trying to use all simulink/matlab capabilities and features in the model. In this part, a case study of constructing **MED-PF-TVC** solar desalination plant is performed.

The first part: In solar PTC application to desalination, the heated oil could be sent to a boiler heat exchanger (BHX), which would generate the necessary steam by a conventional MED-TVC plant. This technique consists of pump unit to overcome the pressure losses, solar collector field (PTC-LS-3 type) for thermal power, BHX for vapor release and MED-PF-TVC type with 5 effects. The organic HTO across the PTC would transfer its thermal power to the fluid (water) across the BHX unit. The generated motive steam is used to compress part of the vapor generated in the last effect by the steam ejector. The expanded motive steam and the recompressed vapor leaving the steam ejector are directed to and condensed in the first effect. Part of the condensate returns to the boiler, and the other part join the potable water product. The vapor formed in the first effect by boiling is directed to the second effect where it acts as a heat source. Figure (10) shows a schematic diagram of the process units for the proposed technique. The specifications and design parameters for this technique are pin pointed as following:

- It is estimated that the daily average global radiation in a typical day in winter would be in the range of 21~22MJ/m^2. To dominate long operation along the day light, the solar radiation would be estimated and fixed at 252W/m^2 (21.4MJ/m^2≈503.7W/m^2 hourly average ≈252W/m^2 daily average). However, under summer conditions it will be expected that there is an excessive energy due to large field area and it might be handled through bypassing some loops in the solar field for maintenance operation.
- Ambient temperature would be set as 20^{o}C.

- The distillate product is fixed at 4545m^3/day (52.6kg/s), and the inlet seawater feed temperature stream is fixed at 30°C with a salinity about 46,000ppm. The outlet brine stream temperature is adjusted as 48.6°C and the number of effects is fixed at 4 effects and the blow down brine salinity is fixed at 69,000ppm.
- According to the design and operating temperature for the recommended HTO and the PTC, the outlet collector temperature would be fixed at 350°C. The motive steam pressure is assigned to be 2500kPa.
- The steam ejector compression ratio is fixed at 2.165.
- The efficiency of the positive displacement pumps are about 75%.
- Effectiveness of all condensers (BHX and MED end condenser) are assigned at 0.8
- PTC configuration and design specifications are adjusted according to LS-3 type.

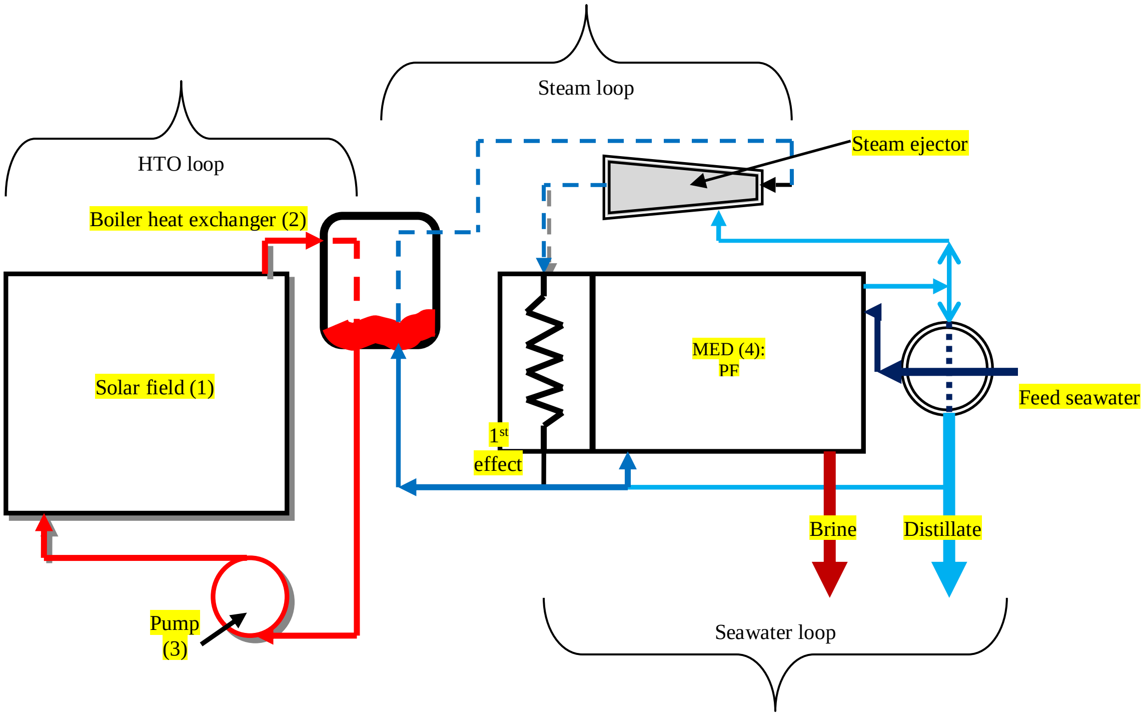
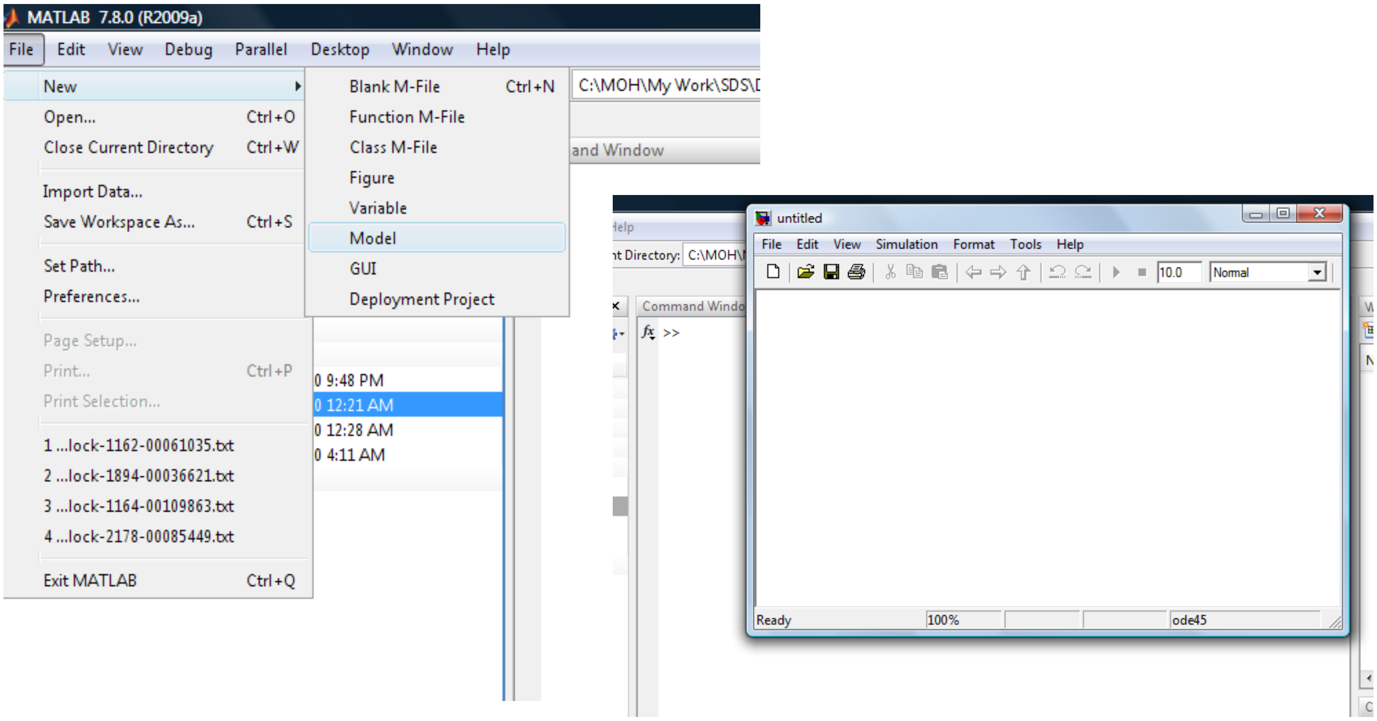


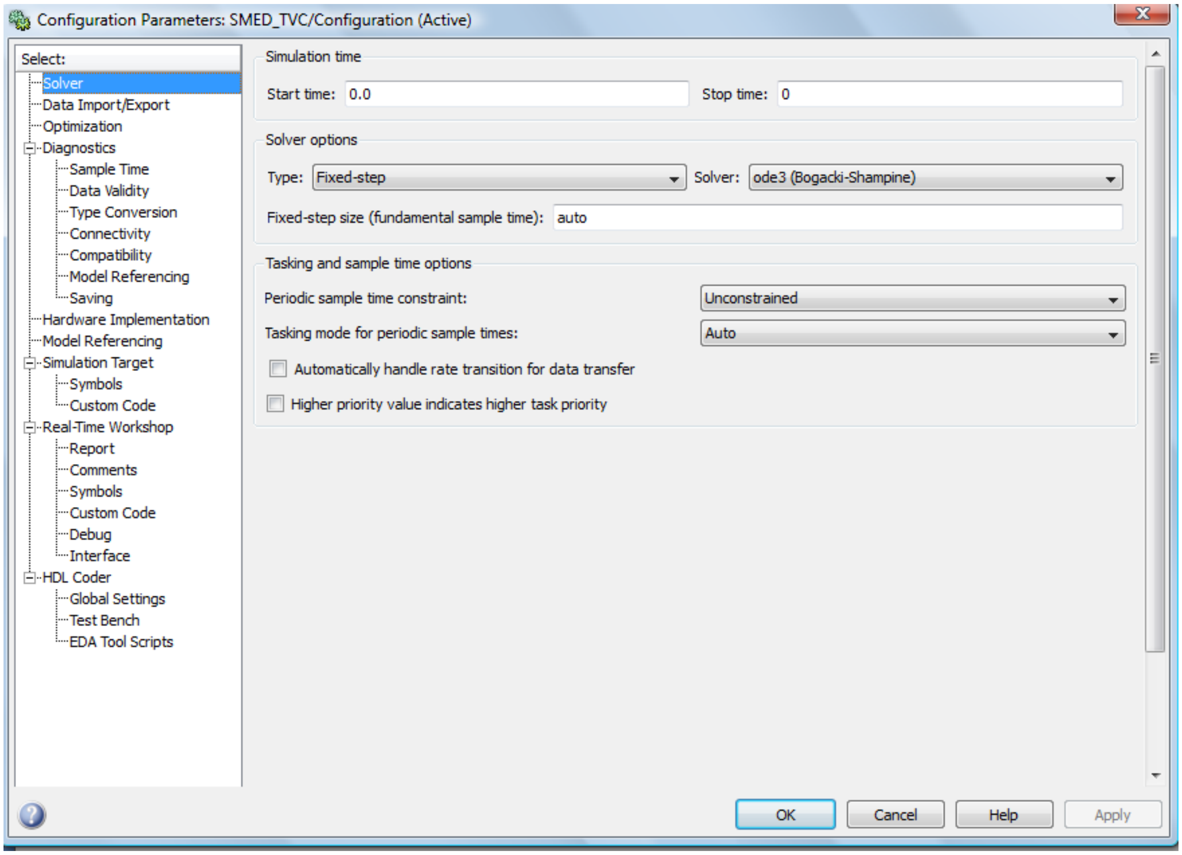
Figure (10) A schematic diagram of solar MED-PF-TVC components: (1) Solar field, (2) Boiler heat exchanger, (3) HTO pump, (4) MED-PF-TVC.

The second part: Based on the first part, the second part (simulink model) could be easily performed. User should do the following steps:

1) From MatLab main menu, open a new simulink browser model by clicking on File>New>Model. A new model window will appear where the user can construct his model.
2) In the untitled new model window user should save the model by clicking on File>save>SMED_TVC.mdl.

3) To adjust your own model according to the type of modeling, click on Simulation>Configuration Parameters or Ctrl-E.
4) Then be sure that the configuration parameters menu is adjusted as follows: Solver type-Fixed step, Solver-od3, Data Import/Export-Uncheck all boxes except Signal logging and Limit data points, Diagnostics-make all none or make it the same as the figure below. Then click Apply and Ok.
5) From the model bar, click on simulation time stop (10sec) and make it 0sec. Then click on the Library Browser button to open the library browser menu.
6) From the Library browser, user should click on User-defined Function button. The selection allow the user to make his own model regardless the already library units that stored in the simulink software. Then user would choose Embedded-MatLab Function block and drag it in the empty model space.
7) Now, user can simulate and arrange any mathematical model inside the embedded block. Double clicking on the embedded block allows the user to write his own mathematical model by specifying the inputs/outputs that he should deal with.
8) User can easily rename the block by just clicking on the block title then change the block name. For solar PTC, let it be “PTC Therminol-VP1”. Inside the block, the mathematical model related to the block is existed.





Configuration Parameters: SMED_TVC/Configuration (Active)

Simulation time

Start time: 0.0 Stop time: 0

Solver options

Type: Fixed-step Solver: ode3 (Bogacki-Shampine)

Fixed-step size (fundamental sample time): auto

Tasking and sample time options

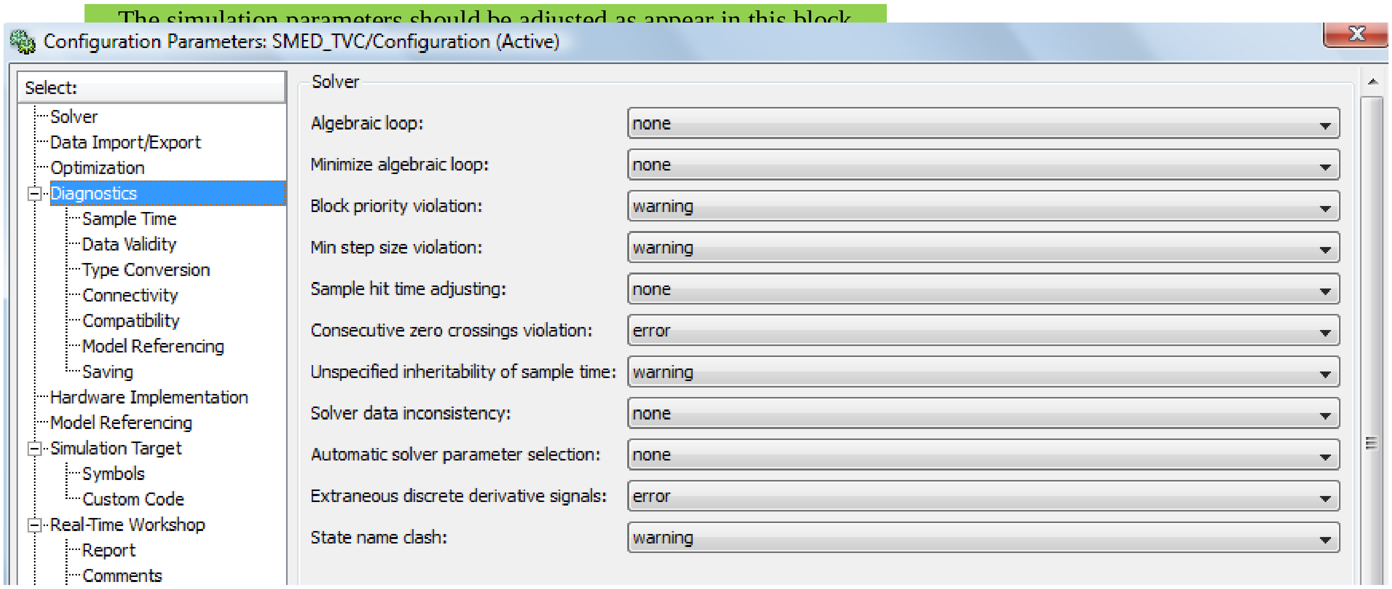
Periodic sample time constraint: Unconstrained

Tasking mode for periodic sample times: Auto

☐ Automatically handle rate transition for data transfer

☐ Higher priority value indicates higher task priority

The simulation parameters should be adjusted as appear in this block



Configuration Parameters: SMED_TVC/Configuration (Active)

Solver

Algebraic loop: none

Minimize algebraic loop: none

Block priority violation: warning

Min step size violation: warning

Sample hit time adjusting: none

Consecutive zero crossings violation: error

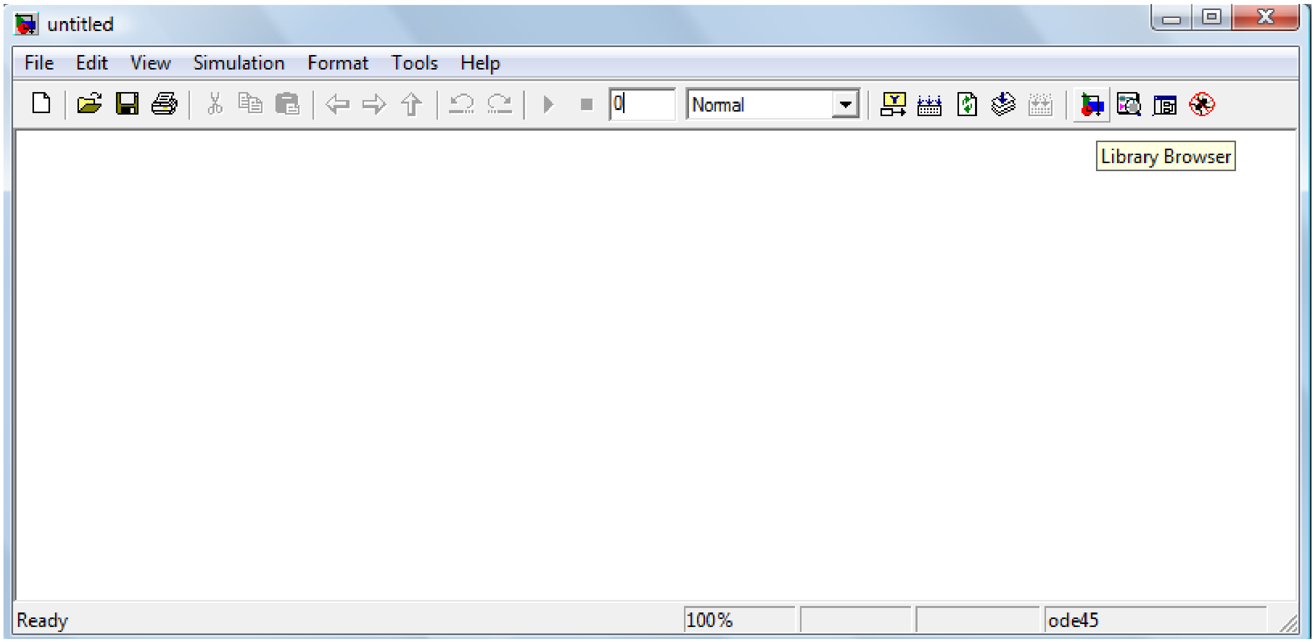
Unspecified inheritability of sample time: warning

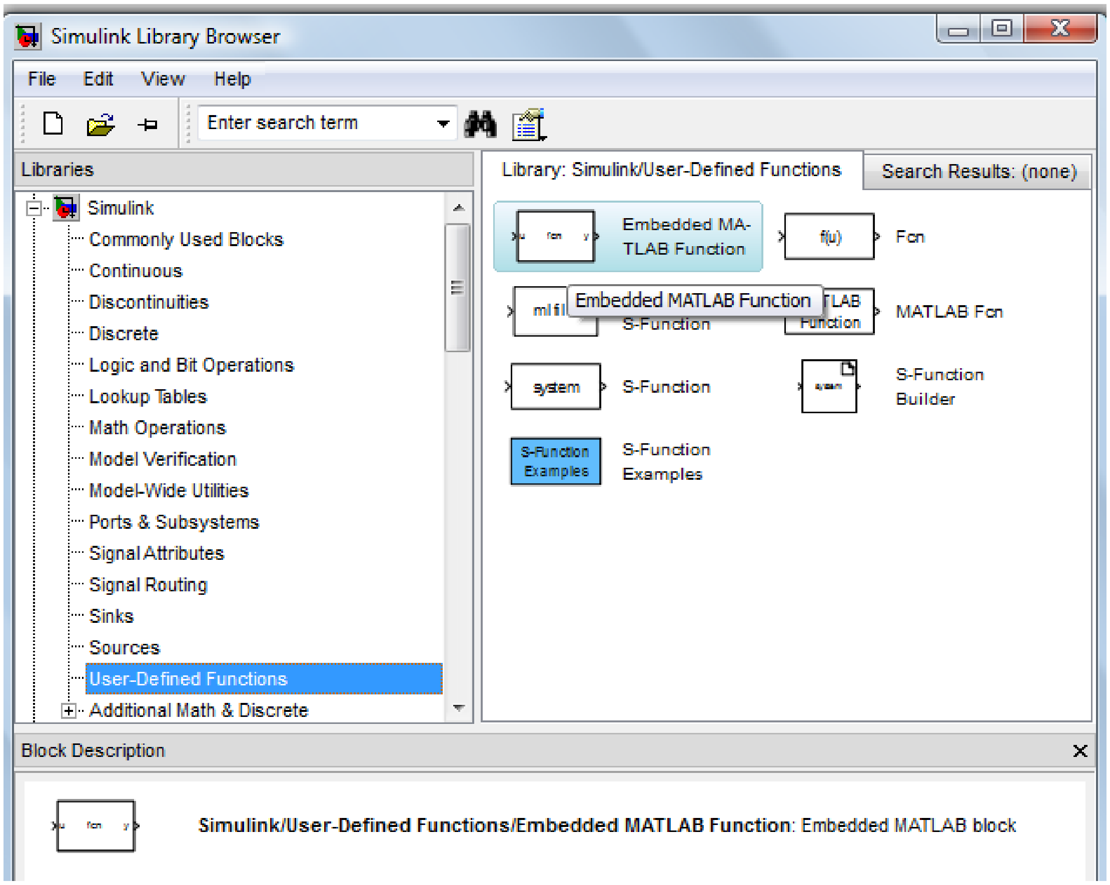
Solver data inconsistency: none

Automatic solver parameter selection: none

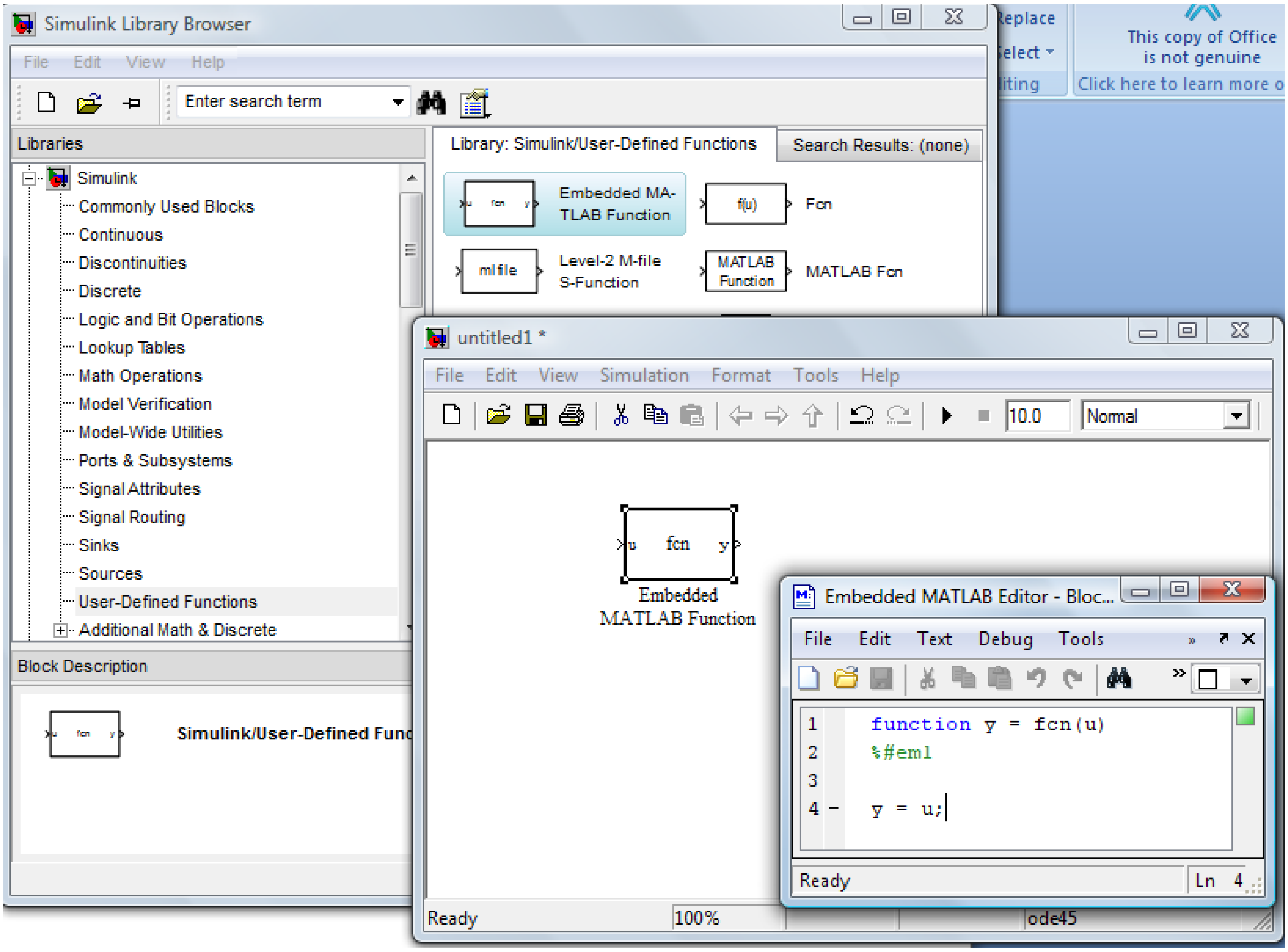
Extraneous discrete derivative signals: error

State name clash: warning

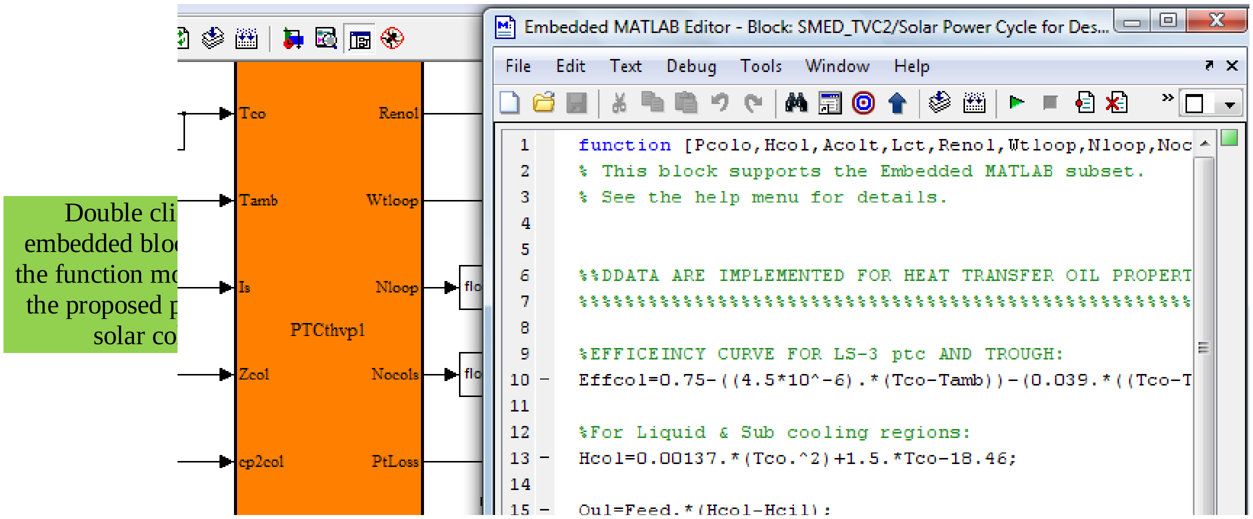




Simulink/User-Defined Functions/Embedded MATLAB Function: Embedded MATLAB block



```
function y = fcn(u)
%#eml

y = u;
```


Tco
Renol
Tamb
Wtloop
Is
Nloop
PTCthvp1
Zcol
Nocols
cp2col
PtLoss
flo
flo
Embedded MATLAB Editor - Block: SMED_TVC2/Solar Power Cycle for Des...
File Edit Text Debug Tools Window Help

Double cli
embedded blo
the function m
the proposed p
solar co

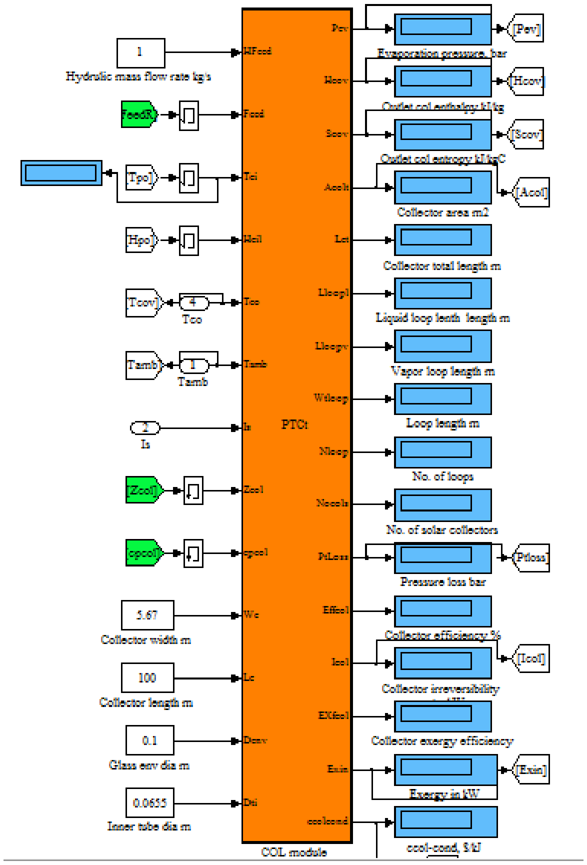
```
function [Pcolo,Hcol,Acolt,Lct,Renol,Wtloop,Nloop,Noc
% This block supports the Embedded MATLAB subset.
% See the help menu for details.


%%DDATA ARE IMPLEMENTED FOR HEAT TRANSFER OIL PROPERT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%EFFICEINCY CURVE FOR LS-3 ptc AND TROUGH:
Effcol=0.75-((4.5*10^-6).*(Tco-Tamb))-(0.039.*((Tco-T

%For Liquid & Sub cooling regions:
Hcol=0.00137.*(Tco.^2)+1.5.*Tco-18.46;

Qul=Feed.*(Hcol-Hcil);
```

The code that governs the relations of **PTC-LS3** solar collector type is obtained as follows:

```
Function
[Pcolo,Hcol,Acolt,Lct,Renol,Wtloop,Nloop,Nocols,PtLoss,Effcol,Icol,EXfcol,Exin,ccolev]=PTCthvp1(HFeed,Feed,Tci,Hcil,Tco,Tamb,Is,Zcol,cp2col,Wc,Lc,Denv,Dti,Klt)

%%DATA ARE IMPLEMENTED FOR HEAT TRANSFER OIL PROPERTIES (HTO) THERMINOL-VP1 (0-427oC):
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%EFFICEINCY CURVE FOR LS-3 ptc AND TROUGH:
Effcol=0.75-((4.5*10^-6).*(Tco-Tamb))-(0.039.*((Tco-Tamb)./Is))-((3*10^-4).*((Tco-Tamb)./Is).^2);

%For Liquid & Sub cooling regions:
Hcol=0.00137.*(Tco.^2)+1.5.*Tco-18.46;

Qul=Feed.*(Hcol-Hcil);

%The aperature area Aa:
Acoll=(1000.*Qul)./(Effcol.*Is);
Acolt=Acoll+0+0;

%collector length:
Lcl=Acoll./(Wc-Denv);
Lct=Lcl+0;

%Solar Field design:
%%%%%%%%%%%%%%%%%%%%
Nloop=Feed./HFeed;
Aloopl=Acoll./Nloop;
Aloop=Acolt./Nloop;
Wtloop=Aloop./Lc;
Lloopl=Aloopl./(Wc-Denv);
Nocols=Acolt./(Lc.*(Wc-Denv));

%Inlet turbine pressure bar:
Pcolo=1.059e-009.*(Tco.^4)-3.412e-007.*(Tco.^3)+3.867e-005.*(Tco.^2)-0.001491.*Tco+0.01249;

%%Presure loss in tubes:
%1:Liquid phase:
Rawl=-6.944.*exp(0.007164.*Tci)+1089.*exp(-0.0007214.*Tci);
dvisl=(6.294.*exp(-0.03614.*Tci)+1.489.*exp(-0.006366.*Tci)).*10^-3;
Renol=(4.*HFeed)./(dvisl.*pi.*Dti);
fl=((1.82.*log(Renol))-1.64).^-2;
DeltaPl=(32.*fl.*Lloopl.*(HFeed.^2))./((Rawl./1).*(pi.^2).*(Dti.^5));

%2:Losses in tubes:
DeltaPtl=Klt.*((8.*(HFeed.^2))./((Rawl./1).*(pi.^2).*(Dti.^4)));

%Total pressure losses:
PtLoss=Nloop.*((DeltaPl+0+DeltaPtl+0).*(10^-5));
% PtLoss=1.*((DeltaPl+0+DeltaPtl+0).*(10^-5));

%Collector Irreversibility:
Cpci=-0.6622*exp(0.001186.*Tci)+2.178*exp(0.0007637.*Tci);
Cpco=-0.6622*exp(0.001186.*Tco)+2.178*exp(0.0007637.*Tco);

Excolo=Feed.*(Hcol-(Cpco.*(Tamb+273).*log((Tco+273)./(Tamb+273))));
Excoli=Feed.*(Hcil-(Cpci.*(Tamb+273).*log((Tci+273)./(Tamb+273))));
Exin=Acolt.*(Is./1000).*(1-((Tamb+273)./(6000)));
Icol=Exin+Excoli-Excolo;
EXfcol=Excolo./(Exin+Excoli);

%Thermoeconomic:
ccolev=((cp2col.*Excoli)+(Zcol./3600))./Excolo;
```

The function line is distinguished as:
Yellow highlight=outputs
Red highlight=function name
Green highlight=inputs

```
Function [outputs]=function_name(inputs)
```



Specified

- Solar radiation
- Ambient temperature
- Outlet collector temperature
- Hydraulic mass flow rate
- Collector width (PTC)
- Collector length (PTC)
- Envelope diameter (PTC)
- Inner tube diameter (PTC)

Calculated

- Evaporation pressure
- Outlet thermophysical stream (enthalpy, entropy)
- Total field area
- Total field width
- Total filed length
- Number of loops
- Number of solar collectors
- Collector energy and exergy efficiencies
- Exergy destruction
- Outlet thermoeconomic streams

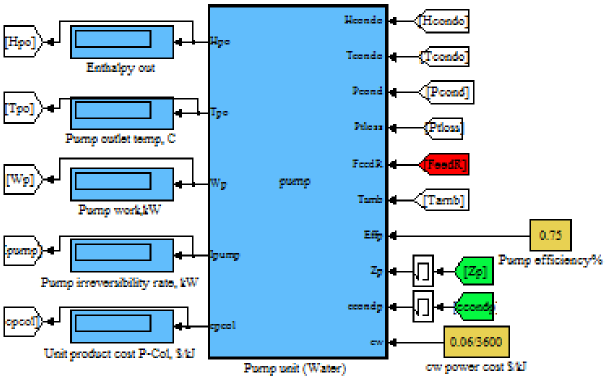
Library blocks used:

1. Constant block
2. User-Defined block
3. SimuLink/Discrete/Memory block
4. SimuLink /Sources/Input
5. SimuLink /Sinks/Display
6. SimuLink /Signal Routing/From
7. SimuLink /Signal Routing/Goto

The recirculation pump (working fluid=Therminol-VP1) code of the solar field is obtained as follows:

```
function [Wp,Tpo,Hpo,Ipump,cpcol]= pump(Ph,Ptloss,Pl,Hpi,Tpi,Mcol,Tamb,Zp,cevp2,cw,Effp)

%Density of HTO kg/m3:
Rawo=-6.944*exp(0.007164.*Tpi)+1089*exp(-0.0007214.*Tpi);

%Pump work kW:
Wp=(100.*Mcol.*(Ph-Pl+Ptloss))./(Rawo.*Effp);
Hpo=(Wp./Mcol)+Hpi;

%outlet temp to collector (HTO):
Tpo=379.2*exp(0.0004137*Hpo)-366.5*exp(-0.001313*Hpo);

%Pump Irreversibility:
Cppi=-0.6622*exp(0.001186.*Tpi)+2.178*exp(0.0007637.*Tpi);
```

```
Cppo=-0.6622*exp(0.001186.*Tpo)+2.178*exp(0.0007637.*Tpo);

Expo=Mcol.*(Hpo-(Cppo.*(Tamb+273).*log((Tpo+273)./(Tamb+273))));
Expi=Mcol.*(Hpi-(Cppi.*(Tamb+273).*log((Tpi+273)./(Tamb+273))));
Ipump=Expi+Wp-Expo;

%Thermo-economic:
cpcol=((cw.*Wp)+(cevp2.*Expi)+(Zp./3600))./Expo;
```



Specified

- Pump efficiency
- Ambient temperature

For some cases (solar desalination with MSF) the cost of power is specified as input parameter however in case of turbine unit it should be calculated.

Calculated

- Pump work
- Outlet thermophysical stream (enthalpy, entropy, temperature)
- Exergy destruction
- Outlet thermoeconomic streams

Library blocks used:

1. Constant block
2. User-Defined block
3. SimuLink/Discrete/Memory block
4. SimuLink /Sinks/Display
5. SimuLink /Signal Routing/From
6. SimuLink /Signal Routing/Goto

The code for boiler heat exchanger unit is demonstrated as follows:

```
function [Hoo,Too,Hvo,Svo,Tvo,Poo,Mcol,Ahev,Nthev,Lthev,Ptlosshev,Iev,cevcond,cevp2]=
hexevp(efhev,Tamb,Hoi,Toi,Hevi,Tevi,Pms,Morc,Klt,Dti,Dto,Dsh,ccolev,ccondev,Zhev)

%Evaporator saturated steam temperature based on motive steam pressure bar (Water):
Tvo=(145.8.*((Pms./100).^0.1959))-48.26;


%outlet saturated vapor enthalpy (Water):
%enthalpy of dry saturated steam at tank pressure and temperature:
Hvo=-3.078e-018*(Tvo.^9)+4.762e-015*(Tvo.^8)-3.076e-012*(Tvo.^7)+1.074e-009*(Tvo.^6)-2.193e-007*(Tvo.^5)+2.646e-
005*(Tvo.^4)-0.001824*(Tvo.^3)+0.06417*(Tvo.^2)+0.894.*Tvo+2504;

%entropy of dry saturated vapor (Water):
Svo=-4.697e-021*(Tvo.^9)+7.274e-018*(Tvo.^8)-4.706e-015*(Tvo.^7)+1.648e-012*(Tvo.^6)-3.39e-010*(Tvo.^5)+4.2e-
008*(Tvo.^4)-3.27e-006*(Tvo.^3)+0.0002231*(Tvo.^2)-0.02823.*Tvo+9.16;

%evaporator thermal power Qhev based on enthalpy difference (Water):
Qhev=Morc.*(Hvo-Hevi);

%outlet temp of HTO:
Too=Toi-(efhev.*(Toi-Tevi));

%outlet HTO liquid enthalpy:
Hoo=0.00137.*(Too.^2)+(1.5.*Too)-18.46;

%outlet heat evaporator sat pressure for HTO:
Poo=1.059e-009.*(Too.^4)-3.412e-007.*(Too.^3)+3.867e-005.*(Too.^2)-0.001491.*Too+0.01249;

%from energy balance across the heat evaporator between HTO and Toluene:
Mcol=Qhev./(Hoi-Hoo);

%mean temp for HTO:
```

```
Tmo=(Toi+Too)./2;

%The overall heat transfer coefficients in the evaporator is calculated as following:
Uhev=1.9695+(1.2057*(10^-2).*Tmo)-(8.5989*(10^-5).*(Tmo.^2))+(2.5651*(10^-7).*(Tmo.^3));

%Total heat transfer area m2:
LMThev=((Toi-Tvo)-(Too-Tevi))./(log((Toi-Tvo)./(Too-Tevi)));
Ahev=Qhev./(Uhev.*LMThev);

%No. of tubes:
Nthev=(Dsh./Dti).^2;

%Tubes length m:
Lthev=Ahev./(pi.*Dto.*Nthev);

%HTO pressure losses in tubes:
Rawo=-6.944*exp(0.007164.*Tmo)+1089*exp(-0.0007214.*Tmo);
% Vo=(4.*Mcol)./(Rawo.*pi.*(Dti.^2));
dviso=(6.294.*exp(-0.03614.*Tmo)+1.489.*exp(-0.006366.*Tmo)).*10^-3;
Renol=(4.*(Mcol./Nthev))./(dviso.*pi.*Dti);
fl=64./Renol;
% fl=((1.82.*log(Renol))-1.64).^-2;
DeltaPl=(32.*fl.*Lthev.*((Mcol./Nthev).^2))./((Rawo./1).*(pi.^2).*(Dti.^5));

%2:Losses in tubes:
DeltaPtl=Klt.*((8.*((Mcol./Nthev).^2))./((Rawo./1).*(pi.^2).*(Dti.^4)));

%Total pressure losses:
Ptlosshev=(DeltaPl+DeltaPtl).*(10^-5);

%Exergy and thermo-economic analysis:
Sevi=1.6*exp(0.002547.*Tevi)-1.622*exp(-0.007713.*Tevi);
Cpoi=-0.6622*exp(0.001186.*Toi)+2.178*exp(0.0007637.*Toi);
Cpoo=-0.6622*exp(0.001186.*Too)+2.178*exp(0.0007637.*Too);
Exevo=Morc.*(Hvo-((Tamb+273).*Svo));
Exevi=Morc.*(Hevi-((Tamb+273).*Sevi));
Exhtoi=Mcol.*(Hoi-(Cpoi.*(Tamb+273).*(log((Toi+273)./(Tamb+273)))));
Exhtoo=Mcol.*(Hoo-(Cpoo.*(Tamb+273).*(log((Too+273)./(Tamb+273)))));

%Total exergy destruction:
Iev=Exevi+Exhtoi-Exevo-Exhtoo;
%thermo-economic analysis:
cevcond=((ccondev.*Exevi)+(Zhev./3600))./Exevo;
cevp2=ccolev;
```

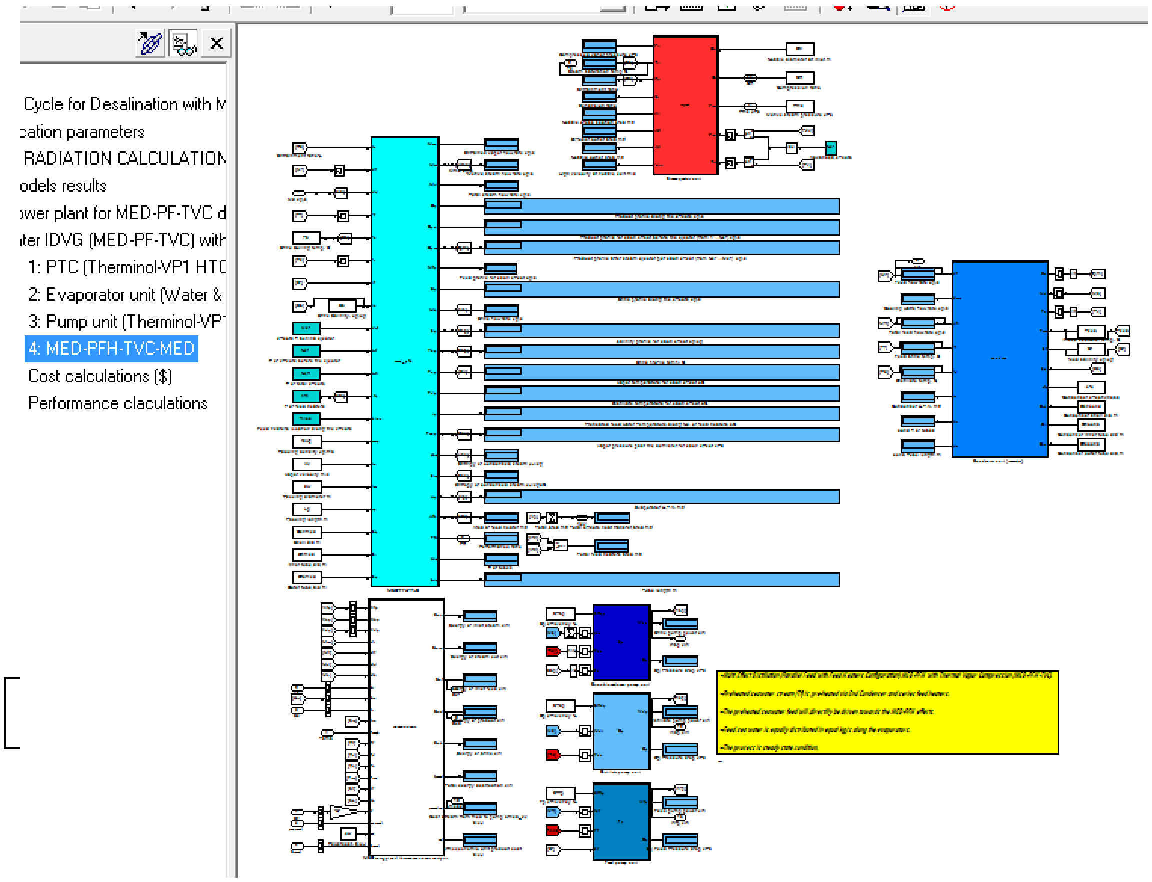
MED-PFH-TVC part is constructed as follows:

1- MED block: contains the mathematical model that arranges the equations related to the MED-PFH process. In this block user can find out the all temperature and mass balance profiles, area profile, and the performance parameters such as gain ratio and other operating conditions.
2- Steam jet ejector block: gives the user the steam relations, areas of the nozzle, velocity of the steam, and the some operating conditions such as pressure and temperature.
3- End condenser block: calculates area, length, mass flow rates, and operating conditions related to the process.
4- Pumps blocks: calculates power, pressure drops.
5- Thermo-economic block: calculated thermo-economic and exergy analysis related to the process.

The user has to specify the following:

- ✓ Efficiency of the pumps.
- ✓ Total number of effects should be vector (1:....Neft)

- ✓ Advanced effects before the steam ejector unit. Suppose that # of effects are 6 effects. So the vector should be written as 1:6 or [1 2 3 4 5 6] or [1:6].
- ✓ The # of advanced effects-for example- should be written as 1:4. Therefore, the vector of effects behind the steam ejector should be as 5:6.
- ✓ # of feed heater also can be entered as a vector such as 1:4 feed heaters or just a value such as 4 or 1 or 3 and so on.
- ✓ The allocation of feed heater is just a number represents the position of the effect that the feed heater should be allocated such as (1 means the 1st effect) (2 means the 2nd effect) and so on.
- ✓ Steam ejector inlet diameter should be specified.



Cycle for Desalination with M
cation parameters
RADIATION CALCULATION
odels results
ower plant for MED-PF-TVC d
ter IDVG (MED-PF-TVC) with
1: PTC (Therminol-VP1 HTC
2: Evaporator unit (Water &
3: Pump unit (Therminol-VP
4: MED-PFH-TVC-MED
Cost calculations ($)
Performance claculations

-Multi Effect Distillation Parallel Feed with Feed Heaters Configuration (MED-PFH) with Thermal Vapor Compression (MED-PFH-TVC)
-Preheated seawater stream (Tf) is preheated via End Condenser and series feed heaters.
-The preheated seawater feed will directly be driven towards the MED-PFH effects.
-Feed sea water is equally distributed in equal kg/s along the evaporators.
-The process is steady state condition.

The code that represents the MED-PFH block is introduced as follows:

```
function [Mev,Mm,Mst,Dp,Dpn,Dpm,Mfp,Bp,Mb,Sp,Tbp,Tvp,Tdp,tp,Pevp,Hsc,Ssc,Ap,Afh,PR,Nte,Lte]=
med_pfh(Ra,Mf,Md,Tf,Tb,Ts,Sf,Sb,Mef,Nef,Neft,nfh,fhloc,rawp,Vv,dw,Lp,Dsh,Dti,Dto)
%#eml

%Weight of the salinity percentage:
Sbw=100.*Sb;

%last effect vapor temperature:
Tv=Tb-fbpe(Tb,Sbw);
```

```
%Temperature drops between effects:
dT=(Ts-Tv)./(max(Neft-0));

%Brine flow temperature profile:
Tbp=sort((Tb+((Neft-1).*dT)),'descend');

%Temperature drop between feed heaters:
dt=(Tbp(fhloc)-Tb)./(max(nfh));

%Preheated Temperature profile along the feed heaters:
tp=sort((Tf+(nfh.*dt)),'descend');

%Vapor temperature distribution:
%vapor temperature at last effect Tv:
Tvp=sort((Tv+((Neft-1).*dT)),'descend');
Tv1=max(Tvp);

%The corresponding saturation vapor pressure Pv kPa is obtained as following:
Pvp=-1.335e-017*(Tvp.^9)+1.328e-014*(Tvp.^8)-5.605e-012*(Tvp.^7)+1.3e-009*(Tvp.^6)-1.786e-007*(Tvp.^5)+1.585e-
005*(Tvp.^4)-0.0007782*(Tvp.^3)+0.02466*(Tvp.^2)-0.2679.*Tvp+1.739;
%The pressure drop in the demister kPa/m (based on backing density, vapor velocity, wire diameter) is evaluated
from the correlation below:
Pp=(3.88178.*(rawp.^0.375798).*(Vv.^0.81317).*(dw.^-1.56114147)).*(Lp/1000);
%The vapor pressure past the demister in kPa is then calculated:
Pevp=Pvp-Pp;
%Suppose that distillate temp (saturation) is obtained at saturation pressure at comp outlet condition:
Tdp=(42.6776-(3892.7./((log(Pevp./1000))-9.48654)))-273.15;

%entropy of condensed steam (saturated liquid):
Ssc=1.6*exp(0.002547.*Ts)-1.622*exp(-0.007713.*Ts);

%enthalpy of condensed steam (saturated liquid):
%Outlet enthalpy from the condenser kJ/kg:
Hsc=1.923e-018*(Ts.^9)-2.974e-015*(Ts.^8)+1.921e-012*(Ts.^7)-6.71e-010*(Ts.^6)+1.372e-007*(Ts.^5)-1.662e-
005*(Ts.^4)+0.001151*(Ts.^3)-0.04126*(Ts.^2)+4.799.*Ts-1.907;

%Mass balance and material:
%Distillate profile:
D1=Md./(flh(Tv1).*(sum(1./flh(Tvp(Neft)))));
Dpn=D1.*(flh(Tv1)./flh(Tvp(Nef)));

%Feed profile for each effect:
Mfp=Mf./max(Neft);

%Brine profile before the ejector:
Bpn=Mfp-Dpn;
Mb=Md.*(Sf./(Sb-Sf));

%Salt profile:
Spn=(Sf.*Mfp)./Bpn;

%the motive steam mass flow rate kg/s:
Tm=(max(Tbp)+max(tp))./2;
Mm=((Mfp.*fcp(Tm,Sf).*((max(Tbp))-(max(tp))))+((max(Dpn)).*flh(max(Tvp))))./(flh(Ts).*(1+(1./Ra)));
%the amount of entrained vapor Mev is then calculated from the entrainment ratio:
Mev=Mm./Ra;
%total steam to evaporator:
Mst=Mev+Mm;

%Distillate profile for effects behind the steam ejector:
Dm=Dpn(max(Nef))-Mev;
Dpm=Dm.*(flh(Tvp(max(Nef)))./flh(Tvp(Mef)));

%Brine profile behind the ejector:
Bpm=Mfp-Dpm;
%Salt profile:
```

```
Spm=(Sf.*Mfp)./Bpm;
Sp=horzcat(Spn,Spm);
Bp=horzcat(Bpn,Bpm);
Dp=horzcat(Dpn,Dpm);

%The system performance parameters are calculated as following:
PR=Md./Mm;

%Heat Transfer areas:
%Evaporator h.t.a:
%the evaporator thermal load Qe:
Qe=(Mm+Mev).*flh(Ts);
Ap=Qe./(fue(Tbp).*dT-(fbpe(Tbp,Sp)));


%No. of tubes:
Nte=(Dsh./Dti).^2;

%Tube length:
Lte=Ap./(pi.*Dto.*Nte);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Heat transfer area for feed heaters:
%LMTfh deference:
DT1=max(Tvp)-max(tp);
DT2=max(Tvp)-Tf;
LMTfh=abs((DT1-DT2)./(log(complex(DT1./DT2))));

%Feed heater area:
Afh=(Mf.*fcp(dt,Sf).*dt)./(fue(dt).*LMTfh);
```

The code for steam ejector unit:

```
function [Ps,Ts,Ra,Er,A1,A2,A3,Vne]= sejectt(Dn,Cr,Pms,Pev,Tv)
%#eml

%The pressure of compressed vapor kPa Ps is function of vapor pressure and compression ratio:
Ps=Cr.*Pev;

%The corresponding saturation temp of the total steam is obtained as following:
Ts=(42.6776-(3892.7./((log(Ps./1000))-9.48654)))-273.15;

%Staureated steam temperature based on motive steam pressure bar (Water):
Tms=(145.8.*((Pms./100).^0.1959))-48.26;

%Motive steam enthalpy based on motive steam temperature:
Hms=-3.078e-018*(Tms.^9)+4.762e-015*(Tms.^8)-3.076e-012*(Tms.^7)+1.074e-009*(Tms.^6)-2.193e-007*(Tms.^5)+2.646e-
005*(Tms.^4)-0.001824*(Tms.^3)+0.06417*(Tms.^2)+0.894.*Tms+2504;

%Enthalpy of saturated vapor steam at nozzle outlet:
Hs=2501.689845+1.806916015.*Ts+5.087717*(10^-4).*(Ts.^2)-1.1221*(10^-5).*(Ts.^3);

%The high velocity at Nozzle exit can be calculated:
Vne=sqrt(2000.*(Hms-Hs));

%The entrainment ratio is obtained as following by calculating temp and pressure correction factors:
PCF=(3*(10^-7)).*(Pms.^2)-(0.0009.*Pms)+1.6101;
TCF=(2*10^-8).*(Tv.^2)-(0.0006.*Tv)+1.0047;
Ra=0.296.*((Ps.^1.19)./(Pev.^1.04)).*((Pms./Pev).^0.015).*(PCF./TCF);%Dessouky

%The expansion ratio:
Er=Pms./Pev;

%The nozzle cross section area based on the nozzle diameter Dn:
A1=pi.*(Dn.^2)./4;
```

```
%The area ratios (A1/A3 & A2/A1) and the areas (A3 & A2) of the nozzle outlet and the diffuser:
Ar1=0.34.*(Ps.^1.09).*(Pms.^-1.12).*(Ra.^-0.16);%A1/A3 by Dessouky page 475 Fundamentals of salt water
desalination
Ar2=1.04.*(Ps.^-0.83).*(Pms.^0.86).*(Ra.^-0.12);%A2/A1 by Dessouky page 475 Fundamentals of salt water
desalination

%Nozzle outlet area m2:
A3=A1./Ar1;

%Diffuser area m2:
A2=A1.*Ar2;
```

Feed pump code:

```
function [Wfp,Dp]= Fp(Efffp,Mf,Tf,Sf)
%#eml

%Pressure difference across the brine pump (Bp)in kPa:
Pf=10.17246-0.6167302.*Tf+1.832249.*(10^-2).*(Tf.^2)-1.77376.*(10^-4).*(Tf.^3)+1.47068*(10^-6).*(Tf.^4);
Dp=Pf-0;
Wfp=(100.*Mf.*Dp)./(fraw(Tf,Sf).*Efffp);
```

Distillate pumps code:

```
function [Wdp,Dp]= Dp(Effdp,Mdt,Tdn)
%#eml

%Pressure difference across the brine pump (Bp)in kPa:
Pd=10.17246-0.6167302.*Tdn+1.832249.*(10^-2).*(Tdn.^2)-1.77376.*(10^-4).*(Tdn.^3)+1.47068*(10^-6).*(Tdn.^4);
Dp=Pd-0;
Wdp=(100.*Mdt.*Dp)./(fraw(Tdn,0).*Effdp);
```

Brine pumps code:

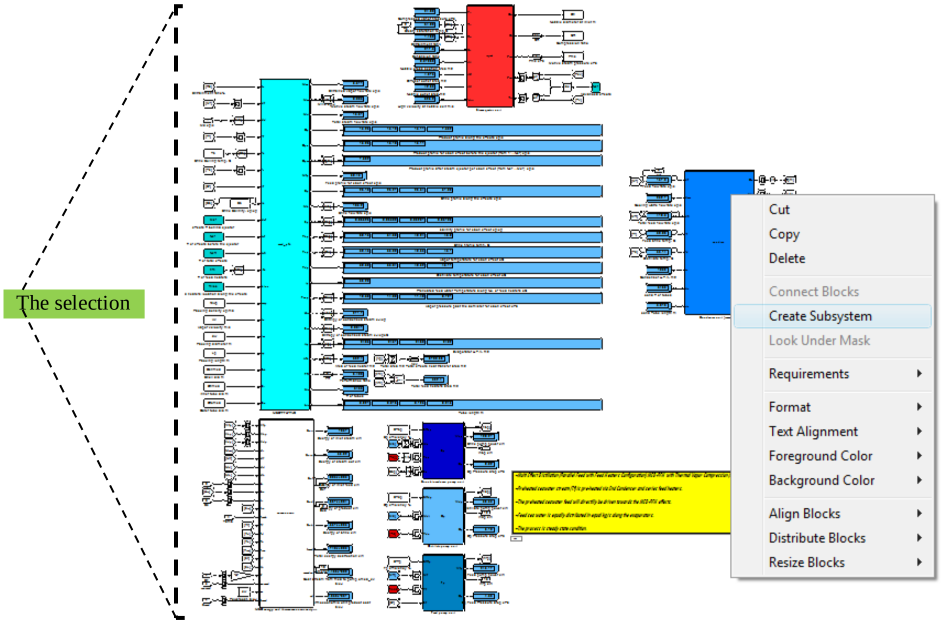
```
function [Wbp,Dp]= Bp(Effbp,Mb,Tbn,Sb)
%#eml

%Pressure difference across the brine pump (Bp)in kPa:
Pb=10.17246-0.6167302.*Tbn+1.832249.*(10^-2).*(Tbn.^2)-1.77376.*(10^-4).*(Tbn.^3)+1.47068*(10^-6).*(Tbn.^4);
Dp=Pb-0;
Wbp=(100.*Mb.*Dp)./(fraw(Tbn,Sb).*Effbp);
```

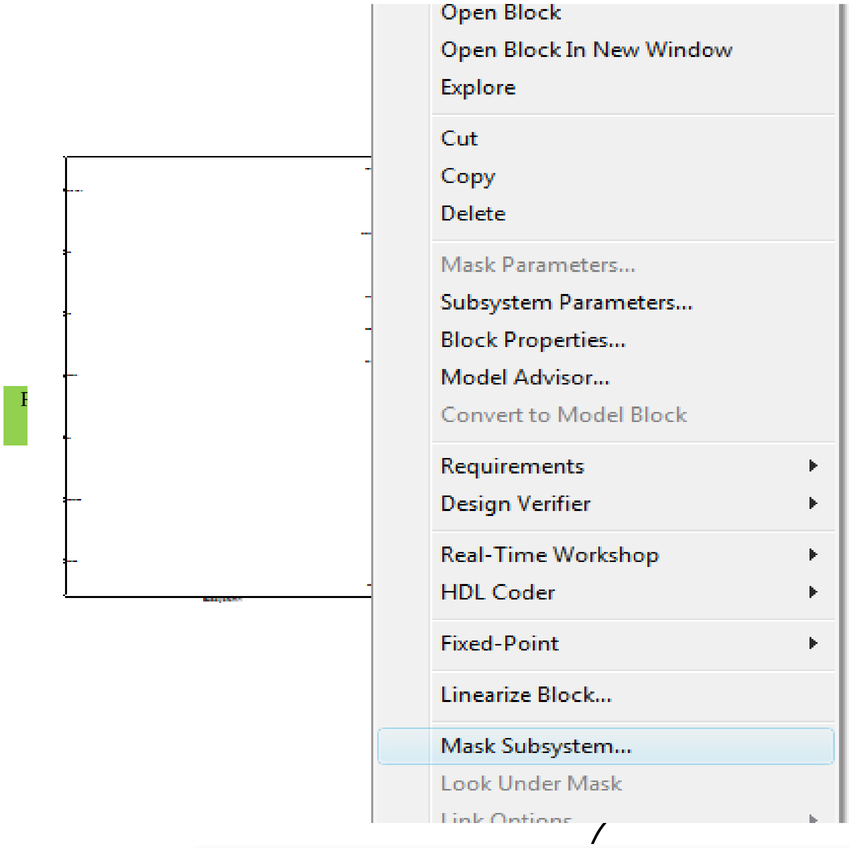
The code for thermo-economic block:

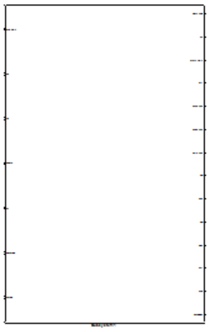
```
function [Exsi,Exso,Exf,Exd,Exb,Imed,cmedev,cd]=
execonomic(Wfp,Wbp,Wdp,Ms,Mf,Md,Mb,Hs,Hsc,Ss,Ssc,Tamb,Tf,Td,Tb,Tsea,Sf,Sb,cf,cevmed,cw,Zmed)
%#eml

Mx=58.8;
Mw=18;
Xb=1000.*Sb;
Xf=1000.*Sf;
Xd=0;
Sd=0;

Xwf=Np(Xf,Mw)./Nmol(Xf,Mw,Mx);
Xsf=Ns(Xf,Mx)./Nmol(Xf,Mw,Mx);

Xwb=Np(Xb,Mw)./Nmol(Xb,Mw,Mx);
Xsb=Ns(Xb,Mx)./Nmol(Xb,Mw,Mx);

Xwd=Np(Xd,Mw)./Nmol(Xd,Mw,Mx);
Xsd=Ns(Xd,Mx)./Nmol(Xd,Mw,Mx);

%Feed stream:
echf=Nmol(Xf,Mw,Mx).*(10^-3).*8.314.*(Tsea+273).*((-Xwf.*(log(Xwf)))-(Xsf.*(log(Xwf))));
```

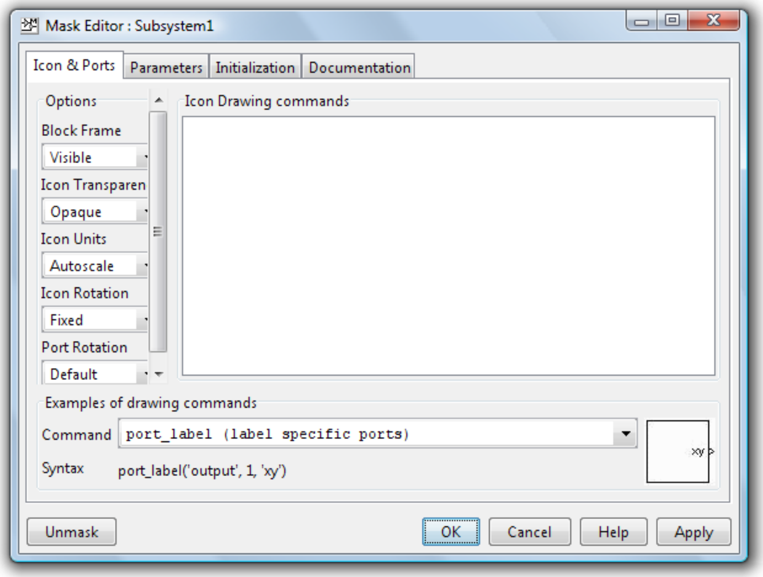
```
ephf=(cp(Tf,Sf).*(Tf+273))-((Tsea+273).*cp(Tf,Sf).*log((Tf+273)./(Tsea+273)));
ef=ephf+echf;
Exf=Mf.*ef;

%distillate stream:
echd=Nmol(Xd,Mw,Mx).*(10^-3).*8.314.*(Tsea+273).*((-Xwd.*(log(Xwd)))-(Xsd.*(log(Xwd))));
ephd=(cp(Td,Sd).*(Td+273))-((Tsea+273).*cp(Td,Sd).*log((Td+273)./(Tsea+273)));
ed=ephd+echd;
Exd=Md.*ed;

%Brine stream:
echb=Nmol(Xb,Mw,Mx).*(10^-3).*8.314.*(Tsea+273).*((-Xwb.*(log(Xwb)))-(Xsb.*(log(Xwb))));
ephb=(cp(Tb,Sb).*(Tb+273))-((Tsea+273).*cp(Tb,Sb).*log((Tb+273)./(Tsea+273)));
eb=ephb+echb;
Exb=Mb.*eb;

%Inlet steam stream:
Exsi=Ms.*(Hs-((Tamb+273).*Ss));

%Outlet steam stream:
Exso=Ms.*(Hsc-((Tamb+273).*Ssc));

%Total exergy destruction:
Imed=Wbp+Wdp+Exsi-Exso+Exf+Exd;

%Thermo-economic analysis:
csi=cevmed;
cso=csi;
cmedev=cso;
cd=((cf.*1)+(cw.*(Wfp+Wbp+Wdp))+(cevmed.*(Exsi-Exso))+(Zmed./3600))./Exd;
```

To mask subsystem, MED-PFH would be masked as an example. The other blocks (PTC, pumps, boiler heat exchanger) should be treated the same. First of all, the desalination plant should be sub-model. After collecting the desalination part as blocks together, user can right click by the mouse choosing “Create Subsystem”.









Block Frame: Visible

Icon Transparen: Opaque

Icon Units: Autoscale

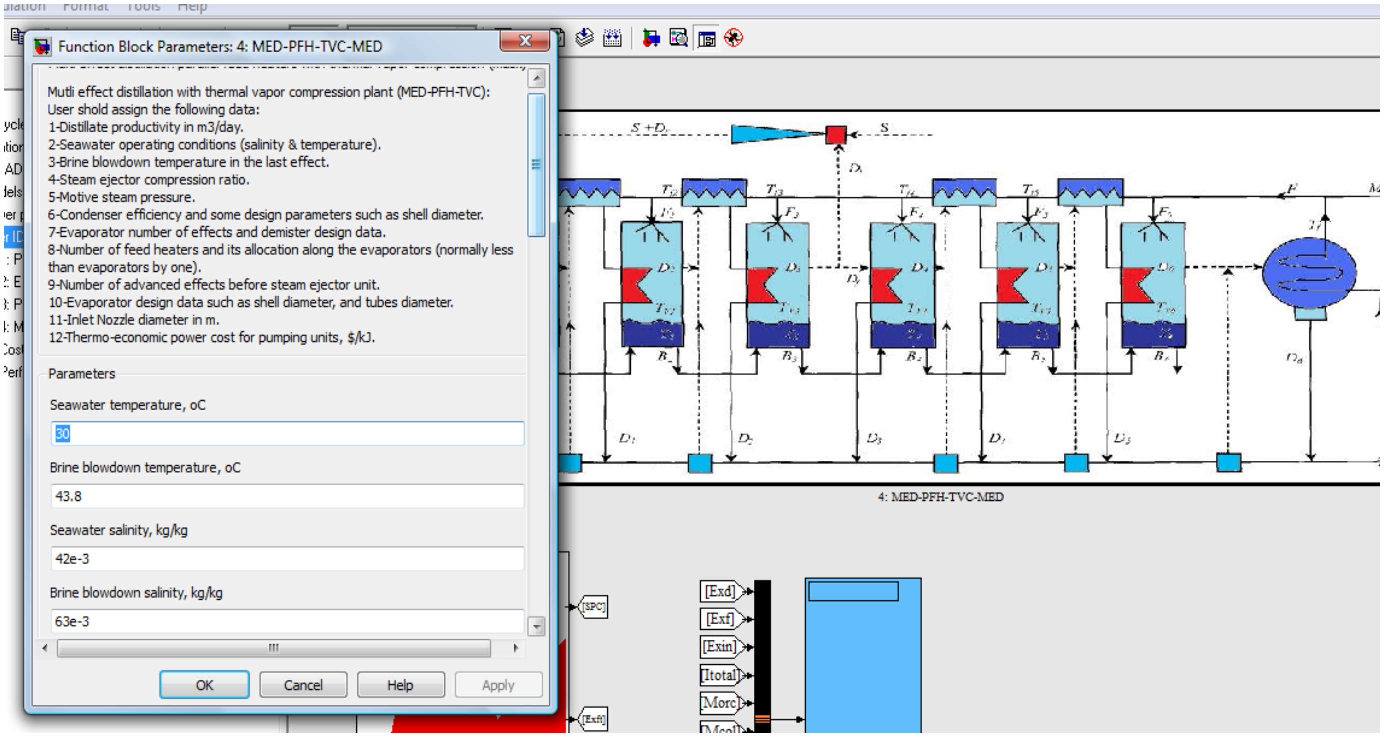
Icon Rotation: Fixed

Port Rotation: Default

Command: `port_label (label specific ports)`

Syntax: port_label('output', 1, 'xy')

By using the mask editor, user can specify the parameters that arrange his model. For the current example, user by double clicking on the sub block can explore the parameters and easily changing and specifying it.



Mutli effect distillation with thermal vapor compression plant (MED-PFH-TVC):
User shold assign the following data:
1-Distillate productivity in m3/day.
2-Seawater operating conditions (salinity & temperature).
3-Brine blowdown temperature in the last effect.
4-Steam ejector compression ratio.
5-Motive steam pressure.
6-Condenser efficiency and some design parameters such as shell diameter.
7-Evaporator number of effects and demister design data.
8-Number of feed heaters and its allocation along the evaporators (normally less than evaporators by one).
9-Number of advanced effects before steam ejector unit.
10-Evaporator design data such as shell diameter, and tubes diameter.
11-Inlet Nozzle diameter in m.
12-Thermo-economic power cost for pumping units, $/kJ.

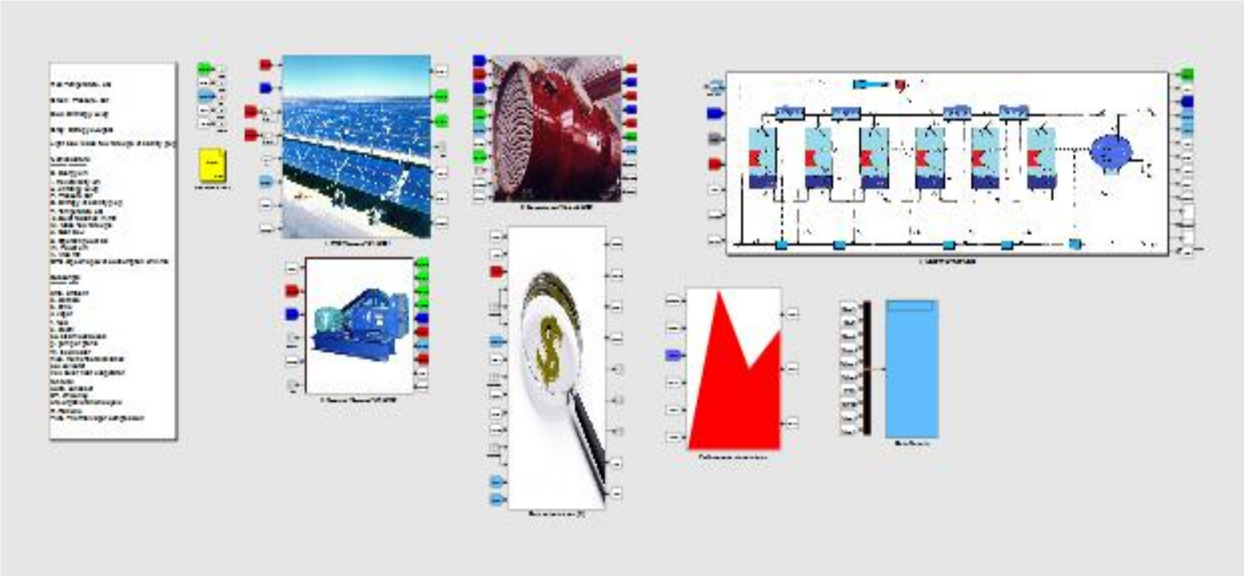
Parameters

Seawater temperature, oC: 30

Brine blowdown temperature, oC: 43.8

Seawater salinity, kg/kg: 42e-3

Brine blowdown salinity, kg/kg: 63e-3



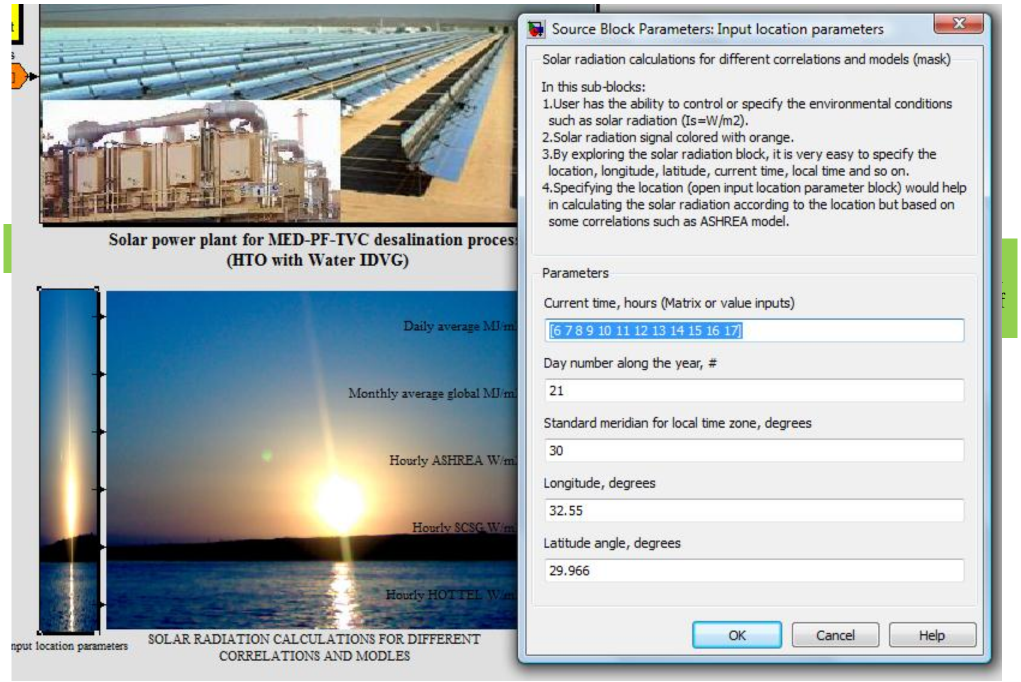
Solar radiation model is presented based on solar angles according to the location of operation. The model sub-system of solar radiation model should appear as in the figure below. In this sub- block:

1. User has the ability to control or specify the environmental conditions such as solar radiation (Is=W/m^2).
2. By exploring the solar radiation block, it is very easy to specify the location, longitude, latitude, current time, local time and so on.
3. Specifying the location (open input location parameter block) would help in calculating the solar radiation according to the location but based on some correlations such as ASHREA and Hottel models.

The cod for such models is presented as follows:

```
function [Iext,Tsun,EoT,ST,tsr,tss,Td,h,Alfa,Zhz,Tht,Azu,wss,delta]= SRAs(CT,n,Lst,Lloc,fai,Beta)

%the extraterrestrial radiation:
Isc=1367;
Iext=Isc.*(1+(0.033.*cos((360.*n.*pi)./(365*180))));

%Mean sun earth distance:
Lse=1.5*10^11;
segma=5.67*10^-8;
%sun diameter:
Rs=(1.39*10^9)/2;
Tsun=((Iext.*(4*pi*(Lse.^2)))./(segma.*(4*pi.*(Rs.^2)))).^0.25;

%solar declination angle:
delta=23.45.*sin(((n+284).*360*pi)./(180*365));

%%solar standard time:
%equation of time:
B=((n-81).*360)./365;
EoT=(9.87.*sin(2.*B*pi/180))-(7.53.*cos(B*pi/180))-(1.5.*sin(B*pi/180));
ST=CT+(EoT./60)-(4.*(Lst-Lloc)./60);

%the hour angle is h:
h=15.*(ST-12);

%the number of hours of day time:
Td=(2*180/(15*pi)).*(acos(-tan(fai.*pi/180).*tan(delta.*pi/180)));

% %hour angle at sun set:
wss=0.5.*Td.*15;

%the solar sun rise and solar sun set:
tsr=12-0.5*Td;
tss=12+0.5*Td;

%%Solar Angles:
%solar altitude angle:
xx=sin(delta.*pi/180).*sin(fai.*pi/180)+cos(delta.*pi/180).*cos(fai.*pi/180).*cos(h.*pi/180);
Alfa=asin(xx).*(180/pi);
function [Ho,Igr,Hmon,Gash,Gscsg,Ghott]= SRMs(Iext,ST,STD,Td,Zhz,wss,fai,delta,AA,BB,CC,Ascsg,Slvl)
%%calculations for daily solar radiation:
Ho=(10^-
6).*((24*3600/pi).*Iext.*((pi.*wss./180).*(sin(fai.*pi/180)).*(sin(delta.*pi/180))+(cos(fai.*pi/180)).*(cos(delt
a.*pi/180)).*(sin(wss.*pi/180))));

%%Solar Radiation Models:
%%%%%%%%%%%%%%%%%%%%%%%%%
%1: ASHREA MODEL:
Gbn=AA.*(exp(-BB./(cos(Zhz.*pi/180))));
Gd=CC.*Gbn;
Gash=(Gbn.*cos(Zhz.*pi/180))+Gd;
%2: SCSG MODEL:
```

```
Gscsg=0.709.*Iext.*(cos(Zhz.*pi./180))-Ascsg;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%3: HOTTEL MODEL:
aoo=0.4237-(0.00812.*(6-Slvl).^2);
a11=0.5055+(0.00595.*(6.5-Slvl).^2);
kk=0.2711+(0.01858.*(2.5-Slvl).^2);

%For tropical climate:
ao=0.95.*aoo;
a1=0.98.*a11;
k=1.02.*kk;
Ghott=Iext.*(ao+(a1.*exp(-k./(cos(Zhz.*pi/180)))));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%THE HOURLY GLOBAL RADIATION RECIVED BY A HORIZONTAL SURFACE:
Igr=Ho.*((pi./(2.*Td)).*(cos((pi/180).*(180.*(ST-12))./Td)));
%CALCULATIONS FOR MONTHLY AVERAGE OF DAILY TOTAL RADIATION ON ORIZONTAL SURFACES:
aa=-0.309+(0.539.*(cos(fai.*pi./180)))-(0.0693.*Slvl)+(0.290.*(STD./Td));
bb=1.527-(1.027.*(cos(fai.*pi./180)))+(0.0926.*Slvl)-(0.359.*(STD./Td));
Hmon=Ho.*(aa+(bb.*(STD./Td)));

%solar zenith angle:
zz=sin(fai.*pi/180).*sin(delta.*pi/180)+cos(fai.*pi/180).*cos(delta.*pi/180).*cos(h.*pi/180);
Zhz=acos(zz).*(180/pi);

%solar incidence angle:
ff=sin(delta.*pi/180)*sin((fai.*pi/180)-(Beta.*pi/180))+cos(Beta.*pi/180)*cos((fai.*pi/180)-
(Beta.*pi/180))*cos(h.*pi/180);
Tht=acos(ff).*(180/pi);

%for solar azimuth angle:
azz=(cos(delta.*pi/180).*sin(h.*pi/180))./cos(Alfa.*pi/180);
Azu=asin(azz).*(180/pi);
```



Source Block Parameters: Input location parameters

Solar radiation calculations for different correlations and models (mask)

In this sub-blocks:
1.User has the ability to control or specify the environmental conditions such as solar radiation (Is=W/m2).
2.Solar radiation signal colored with orange.
3.By exploring the solar radiation block, it is very easy to specify the location, longitude, latitude, current time, local time and so on.
4.Specifying the location (open input location parameter block) would help in calculating the solar radiation according to the location but based on some correlations such as ASHREA model.

Parameters

Current time, hours (Matrix or value inputs): [6 7 8 9 10 11 12 13 14 15 16 17]

Day number along the year, #: 21

Standard meridian for local time zone, degrees: 30

Longitude, degrees: 32.55

Latitude angle, degrees: 29.966

4.5 Data results: **MED-PF-TVC** Case study

Based on the previous studies [(1997) Desalination 110 (1997) 223–238], the following example is evaluated and compared with Darwish [(1997) Desalination 110 (1997) 223–238]. The specified parameters are pin pointed earlier in subsection 4.4. The results obtained revealed that there is a very good agreement between the SDS package and Darwish. The results are obtained based on the following input data:

M_d=4545m^3/d,
T_{sea}=30^{o}C,
T_{bn}=46.8^{o}C,
S_f=46g/kg, S_b=69g/kg,
CR=2.165, Condenser efficiency=0.65, Motive steam pressure=2500kPa

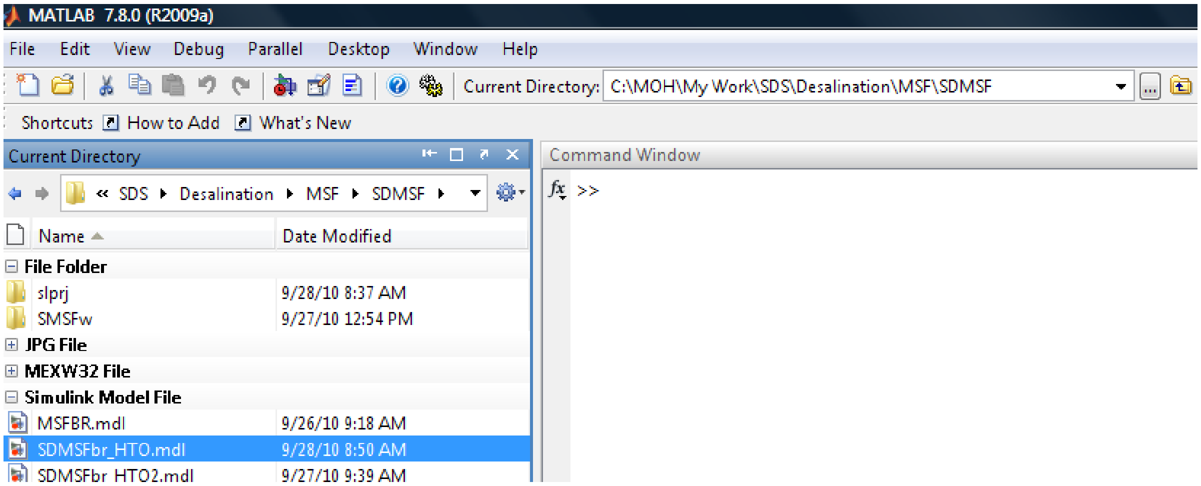
Preheated feed temperature °C		**Distillate temperature from end condenser °C**		**Brine temperature profile °C**		**Vapor temperature profile °C**	
SDS	**Darwish**	**SDS**	**Darwish**	**SDS**	**Darwish**	**SDS**	**Darwish**
40.4	*40	35.53	not assigned	58.83	58	58.048	not mentioned
				54.8	54	54.038	not mentioned
				50.8	50	50.027	not mentioned
				46.8	46	46.016	

Distillate temperature profile °C		**Md profile kg/s**		**Mb profile kg/s**		**Ms total kg/s**		**Mf kg/s**	
SDS	**Darwish**	**SDS**	**Darwish**	**SDS**	**Darwish**	**SDS**	**Darwish**	**SDS**	**Darwish**
57.99	not mentioned	13.23	14	26.221	....	14.57	14.88	157.8	157.86
53.99	not mentioned	13.177	13.2	26.27	....	GOR=6.2	GOR=7.65		
49.99	not mentioned	13.12	12.86	26.329	....				
46.011	not mentioned	13.07	12.56	26.38	....				
		52.597	**52.6**	**105.2**	**....**				

M_{cw} cooling water kg/s		**T_{steam} °C**		**P_{steam} kPa**		**Entrained vapor kg/s**		**Motive steam kg/s**	
SDS	**Darwish**	**SDS**	**Darwish**	**SDS**	**Darwish**	**SDS**	**Darwish**	**SDS**	**Darwish**
185.2	185.42	62.06	62	21.92	21.85	6	6.88	8.6	8

4.6 Multi-Stage Flash Brine Recycle (MSF-BR) Model: Case Study

To run the solar combined with MSF-BR desalination model, user has to log into the SDS path then double clicking on the file name of the model (SDMSFbr_HTO.mdl).

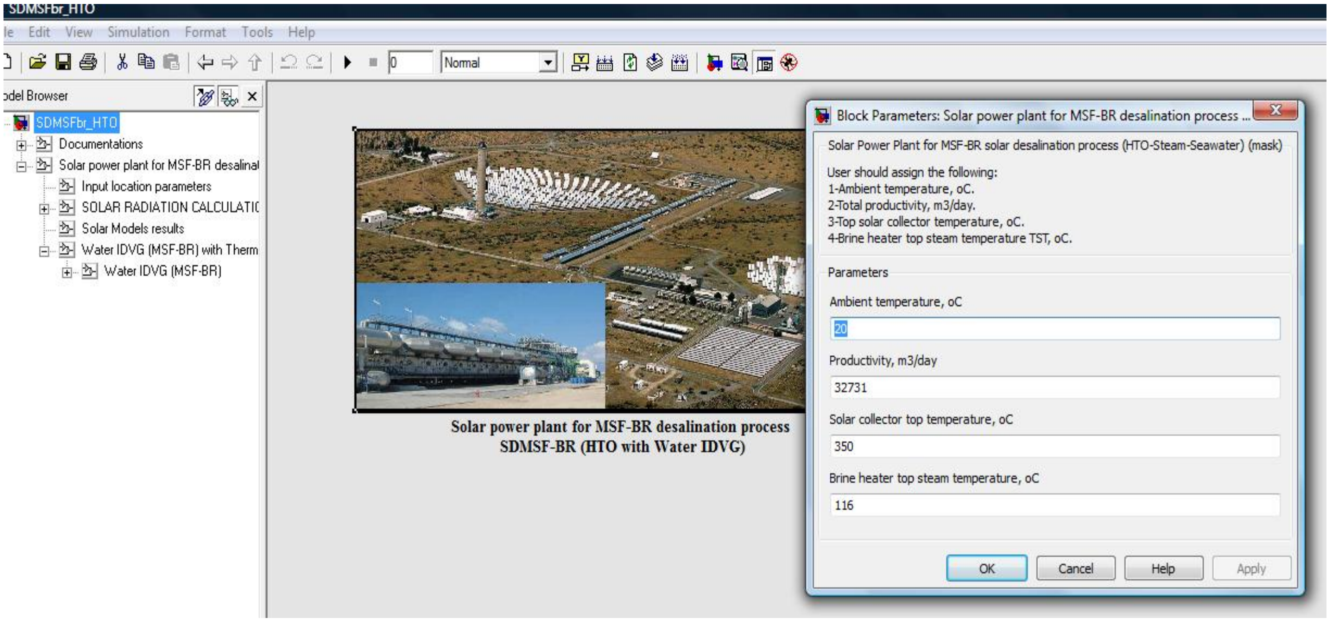


Current Directory: C:\MOH\My Work\SDS\Desalination\MSF\SDMSF

« SDS ▸ Desalination ▸ MSF ▸ SDMSF ▸

Name	Date Modified
File Folder	
slprj	9/28/10 8:37 AM
SMSFw	9/27/10 12:54 PM
JPG File	
MEXW32 File	
Simulink Model File	
MSFBR.mdl	9/26/10 9:18 AM
SDMSFbr_HTO.mdl	9/28/10 8:50 AM
SDMSFbr_HTO2.mdl	9/27/10 9:39 AM

To assign the plant productivity m^3/day, ambient temperature °C and solar radiation data, user has to click on the model explorer icon then double clicking on the main block. The block parameters menu will open and the user can easily assign the main input data. Also it is very easy to assign the data of the location such as longitude, latitude, and the day number of the year.


SDMSFbr_HTO
le
Edit
View
Simulation
Format
Tools
Help
0
Normal
odel Browser
SDMSFbr_HTO
Documentations
Solar power plant for MSF-BR desalinat
Input location parameters
SOLAR RADIATION CALCULATIO
Solar Models results
Water IDVG (MSF-BR) with Therm
Water IDVG (MSF-BR)
Solar power plant for MSF-BR desalination process
SDMSF-BR (HTO with Water IDVG)
Block Parameters: Solar power plant for MSF-BR desalination process ...
OK
Cancel
Help
Apply

Solar Power Plant for MSF-BR solar desalination process (HTO-Steam-Seawater) (mask)

User should assign the following:
1-Ambient temperature, oC.
2-Total productivity, m3/day.
3-Top solar collector temperature, oC.
4-Brine heater top steam temperature TST, oC.

Parameters

Ambient temperature, oC: 20

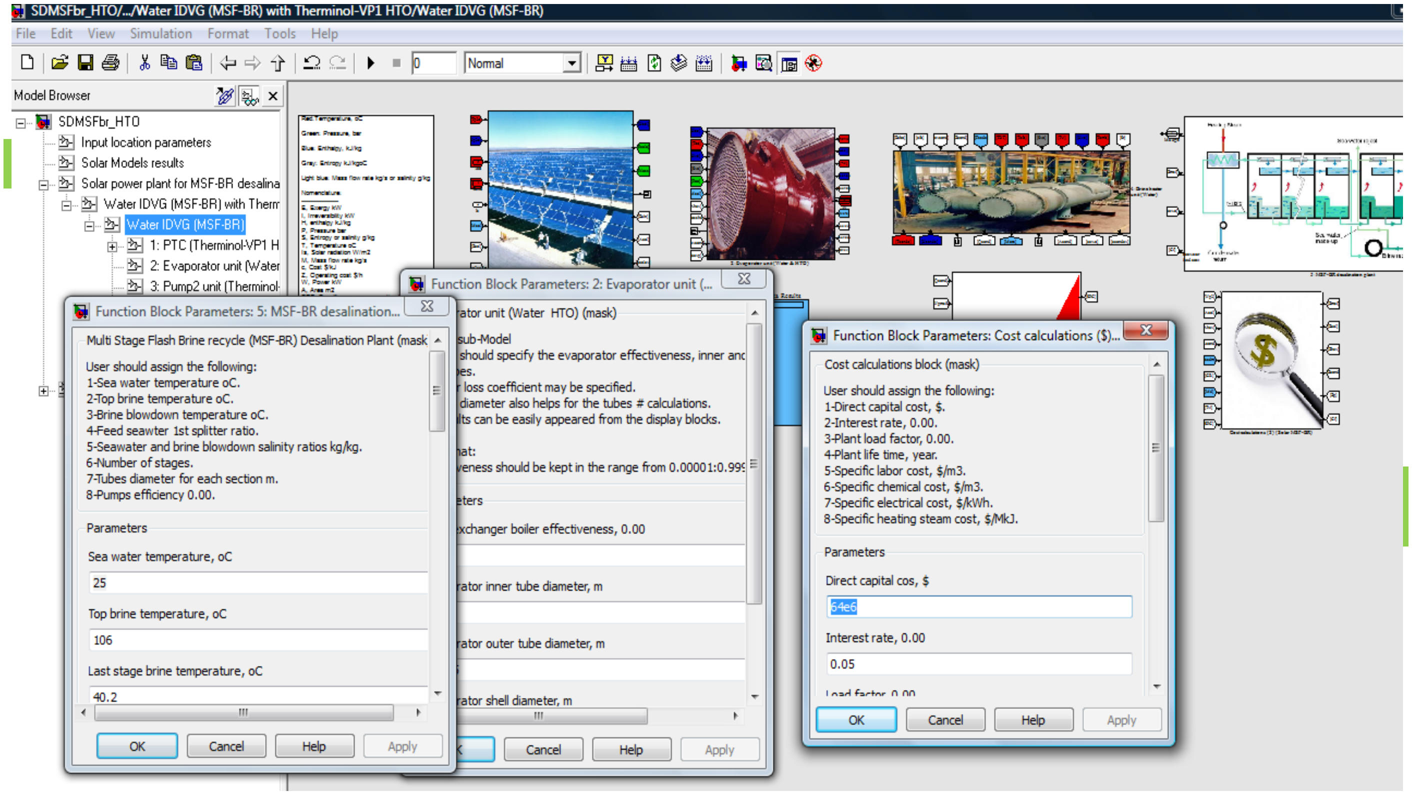
Productivity, m3/day: 32731

Solar collector top temperature, oC: 350

Brine heater top steam temperature, oC: 116

In the sub menu, user can easily find out the units and the sub-units that represent the proposed process. In the following figure, the process units are PTC solar field, boiler heat exchanger unit, pump, brine heater, and MSF-BR desalination plant. By double clicking on the blocks, user can easily specify the operating conditions and the permitted design consideration for each unit individually. User also has the ability to use the capabilities of the MatLab/SimuLink that included in the software tools. The tools are concluded in to the following items:

- ✓ Users can easily copying the units and duplicate and pasting them.
- ✓ User can delete the unwanted units.
- ✓ User can take copies to clipboard with high permeation to edit and reform.
- ✓ Also printing the models and their sub models is easily available.
- ✓ User can redo his work for instant accident such as removing or deleting any parameter or unit.
- ✓ User can drive out his results through different ways such as “mat” files, matlab “workspace”, or/and display block.
- ✓ Also, it is become easy for the user to handle the “mat” to an “excel sheet” or construct a new figure.



SDMSFbr_HTO/.../Water IDVG (MSF-BR) with Therminol-VP1 HTO/Water IDVG (MSF-BR)

Function Block Parameters: 5: MSF-BR desalination...

Multi Stage Flash Brine recycle (MSF-BR) Desalination Plant (mask

User should assign the following:
1-Sea water temperature oC.
2-Top brine temperature oC.
3-Brine blowdown temperature oC.
4-Feed seawter 1st splitter ratio.
5-Seawater and brine blowdown salinity ratios kg/kg.
6-Number of stages.
7-Tubes diameter for each section m.
8-Pumps efficiency 0.00.

Parameters

Sea water temperature, oC
25

Top brine temperature, oC
106

Last stage brine temperature, oC
40.2

Function Block Parameters: 2: Evaporator unit (...

rator unit (Water HTO) (mask)

sub-Model
should specify the evaporator effectiveness, inner and
es.
r loss coefficient may be specified.
diameter also helps for the tubes # calculations.
lts can be easily appeared from the display blocks.

at:
veness should be kept in the range from 0.00001:0.999

eters

xchanger boiler effectiveness, 0.00

rator inner tube diameter, m

rator outer tube diameter, m

rator shell diameter, m

Function Block Parameters: Cost calculations ($)...

Cost calculations block (mask)

User should assign the following:
1-Direct capital cost, $.
2-Interest rate, 0.00.
3-Plant load factor, 0.00.
4-Plant life time, year.
5-Specific labor cost, $/m3.
6-Specific chemical cost, $/m3.
7-Specific electrical cost, $/kWh.
8-Specific heating steam cost, $/MkJ.

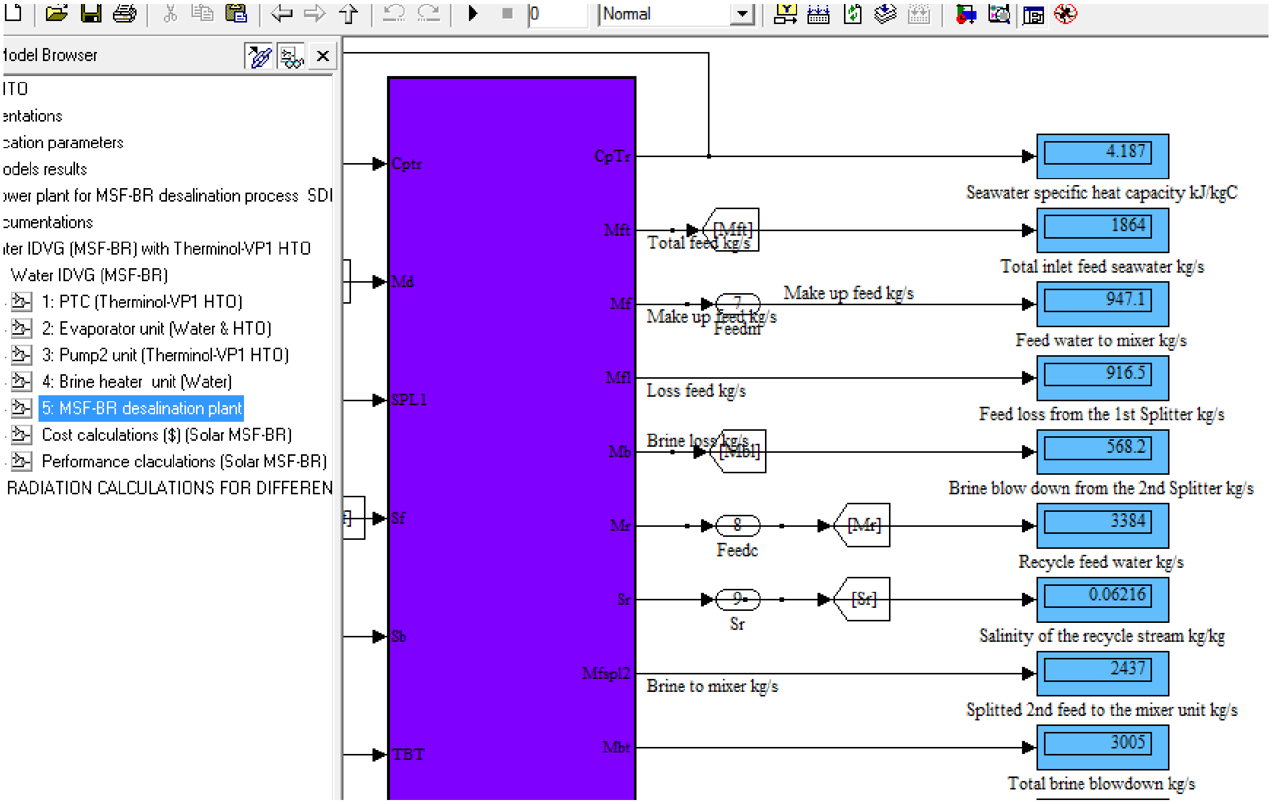
Parameters

Direct capital cos, $
64e6

Interest rate, 0.00
0.05

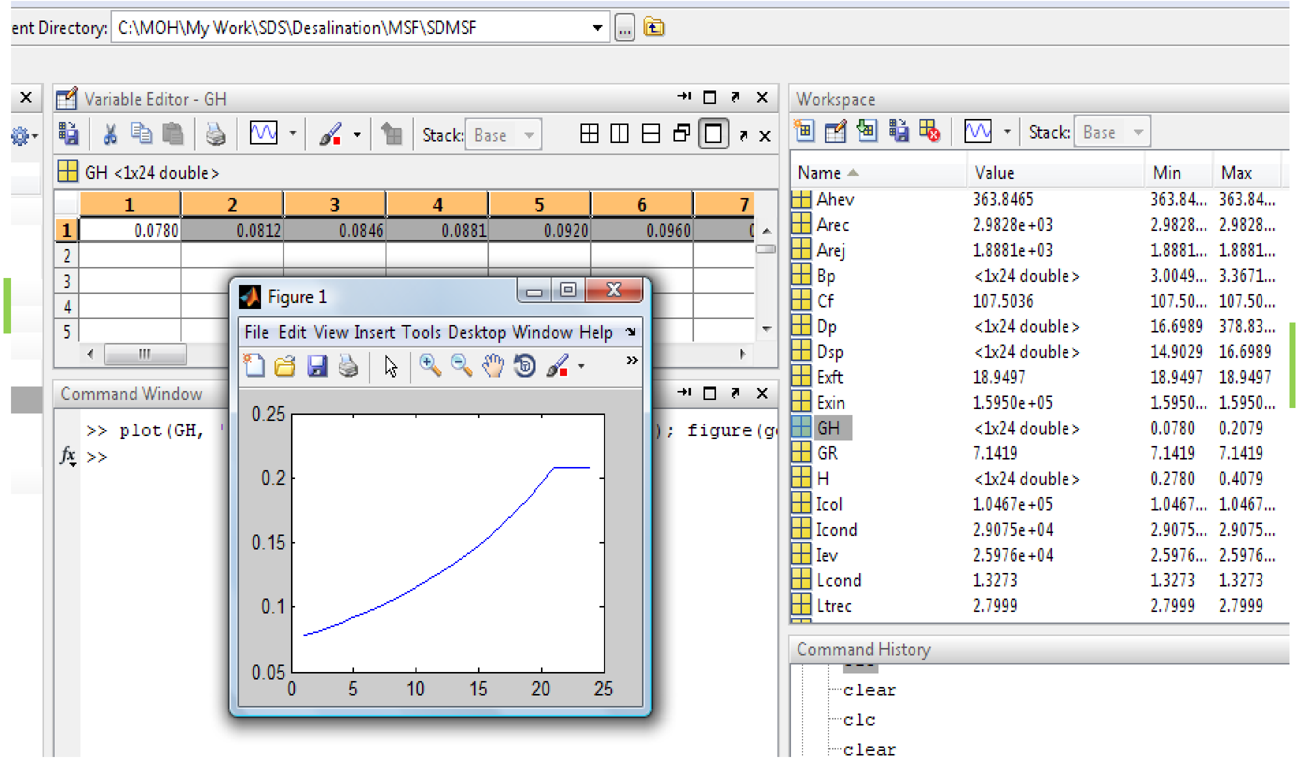
Load factor, 0.00

Specification blocks related to each unit



Model Browser
ITO
entations
cation parameters
odels results
ower plant for MSF-BR desalination process SDI
cumentations
ter IDVG (MSF-BR) with Therminol-VP1 HTO
Water IDVG (MSF-BR)
1: PTC (Therminol-VP1 HTO)
2: Evaporator unit (Water & HTO)
3: Pump2 unit (Therminol-VP1 HTO)
4: Brine heater unit (Water)
5: MSF-BR desalination plant
Cost calculations ($) (Solar MSF-BR)
Performance claculations (Solar MSF-BR)
RADIATION CALCULATIONS FOR DIFFEREN

4.187	Seawater specific heat capacity kJ/kgC
1864	Total inlet feed seawater kg/s
947.1	Feed water to mixer kg/s
916.5	Feed loss from the 1st Splitter kg/s
568.2	Brine blow down from the 2nd Splitter kg/s
3384	Recycle feed water kg/s
0.06216	Salinity of the recycle stream kg/kg
2437	Splitted 2nd feed to the mixer unit kg/s
3005	Total brine blowdown kg/s



ent Directory: C:\MOH\My Work\SDS\Desalination\MSF\SDMSF

GH <1x24 double>

	1	2	3	4	5	6	7
1	0.0780	0.0812	0.0846	0.0881	0.0920	0.0960	0
2							
3							
4							
5							

```
>> plot(GH, '                                   ); figure(g
>>
```

Name	Value	Min	Max
Ahev	363.8465	363.84...	363.84...
Arec	2.9828e+03	2.9828...	2.9828...
Arej	1.8881e+03	1.8881...	1.8881...
Bp	<1x24 double>	3.0049...	3.3671...
Cf	107.5036	107.50...	107.50...
Dp	<1x24 double>	16.6989	378.83...
Dsp	<1x24 double>	14.9029	16.6989
Exft	18.9497	18.9497	18.9497
Exin	1.5950e+05	1.5950...	1.5950...
GH	<1x24 double>	0.0780	0.2079
GR	7.1419	7.1419	7.1419
H	<1x24 double>	0.2780	0.4079
Icol	1.0467e+05	1.0467...	1.0467...
Icond	2.9075e+04	2.9075...	2.9075...
Iev	2.5976e+04	2.5976...	2.5976...
Lcond	1.3273	1.3273	1.3273
Ltrec	2.7999	2.7999	2.7999

```
clear
clc
clear
```

Consider an example of MSF-BR desalination plant with a capacity of **32728m^3/day**. The input parameters and specifications are illustrated in the following table. The process validity of MSF-BR example is examined with Dessouky (Fundamental of salt water desalination, Book) and also illustrated in Table below. The data results show a good agreement for the developed program (SDS) with Dessouky results.

32728m^3/day MSF-BR

Top brine temperature (TBT), °C	106
Brine blow down temperature, °C	40.2
Feed seawater temperature, °C	25
Cooling water splitter ratio	0.5082
Sea water salinity, ppm	42000
Brine blow down salinity, ppm	70000
Tube outside diameter (brine heater), m	0.02199
Tube outside diameter (heat rejection), m	0.0239
Tube outside diameter (heat recovery), m	0.01299
No. of stages	24 (21/3)
Recirculation pump efficiency, %	75
Chamber Load, kg/s.m	180
Vapor velocity, m/s	12
Weir coefficient	0.5

Design point results:	**Dessouky**	**SDS**
Total feed, kg/s	1861	1862
Distillate flow rate, kg/s	378.8	378.8
Make up, kg/s	947	947
Brine recycle flow rate, kg/s	3384	3384
Brine blow-down flow rate, kg/s	568.2	568.2
Steam mass flow rate, kg/s	52.52	53.04
Cooling water splitter ratio	0.5087	0.5087
Top vapor temperature, °C	101.2	101.98
Top brine temperature (TBT), °C	110	110
Top feed temperature, °C	97.75	97.78
Recycle blow down temperature, °C	48.25	48.42
Vapor temperature at last stage, °C	--	38.2
Recycle stream salinity, ppm	62170	62163
1st stage gate height, m	0.078	0.0779
1st stage brine pool height, m	0.278	0.2779
Stage length, m	2.56	2.58
Performance ratio	7.21	7.14

The code that represents the MSF-BR is presented as follows:

```
function
[CpTr,Mft,Mf,Mfl,Mb,Mr,Sr,Mfspl2,Mbt,SPL2,Bp,Dsp,Dp,Srp,Tbp,Tbprec,Tvp,Trprec,Tfprej,Tbprej,Arec,Arej,GH,H,Wstg,
Lstg,Ltrec,Ltrej,Ntrec,Ntrej]=
msf_br(Cptr,Md,SPL1,Sf,Sb,TBT,Tb,Tsea,Nt,Nhr,Nrej,Vb,Vv,Cd,Dtohr,Dtor,Dtihr,Dtir,Vf)

%Feed stream to the mixer unit:
Mf=(Sb./(Sb-Sf)).*Md;
```

```
%Total needed feed (feed loss from the 1st mixer+feed goes to the mixer):
Mft=Mf./SPL1; SPL1 is the 1st splitter unit

%Feed loss from the 1st splitter:
Mfl=Mft-Mf;

%The brine loss blow-down:
Mb=Mf-Md;

%Stage temp drop:
Tstg=(TBT-Tb)./(max(Nt));
```

Where TBT is the top brine temperature, Tb is the last stage temperature, and Nt the total number of stages.

```
%Average temperature:
Tav=(TBT+Tb)./2;
```

The average temperature is used to calculate Cp and latent heat therefore; the recycle stream is calculated based on the following relation

```
%The recycle brine flow rate:
Y=(Cptr.*Tstg)./fLH(Tav);
Mr=Md./(1-((1-Y).^(max(Nt))));
```

And the recycle stream salinity is then obtained

```
Sr=((Sf.*Mf)+((Mr-Md).*Sb)-(Mb.*Sb))./Mr;
```

The following relation is used to draw the distillate profile per each stage

```
%Stage distillate/stage profile:
Dsp=Mr.*Y.*(1-Y).^(Nt-1);
```

From the calculation of the recycle stream it is become easy to calculate the feed stream from the 2nd splitter and total brine and the 2nd splitter ratio.

```
%Feed stream from spl2:
Mfspl2=Mr-Mf;

%Total brine:
Mbt=Mfspl2+Mb;

%2nd Spliiter ratio:
SPL2=Mfspl2./Mbt;
```

From distillate profile, brine profile can be drawn from the following relation

```
%Brine flow rate & feed profiles:
Bp=Mr-Dp;
```

And the salinity profile can be drawn

```
%The salinity profile:
Srp=((sort(Bp,'ascend')).*Sb)./Mr;

%temperature profile for heat recovery section:
%Brine temperature profile:
Tbprec=TBT-(Tstg.*Nhr);

%Recycle temperature profile:
Trprec=Tb+((sort(Nhr,'descend')).*Tstg);

%Mean recycle temperature profile:
Trr=mean(Trprec);

%Calculated specific heat capacity based on mean recycle temperature and salinity:
CpTr=fcp(Trr,Sr);

%Temperature profile of the heat rejection stages:
%Stage temperature drop:
Tstgrej=(Tb-Tsea)./(max(Nrej));

%Feed profile for rejection section:
Tfprej=Tsea+(((max(Nrej))-(Nrej-1)).*Tstgrej);
```

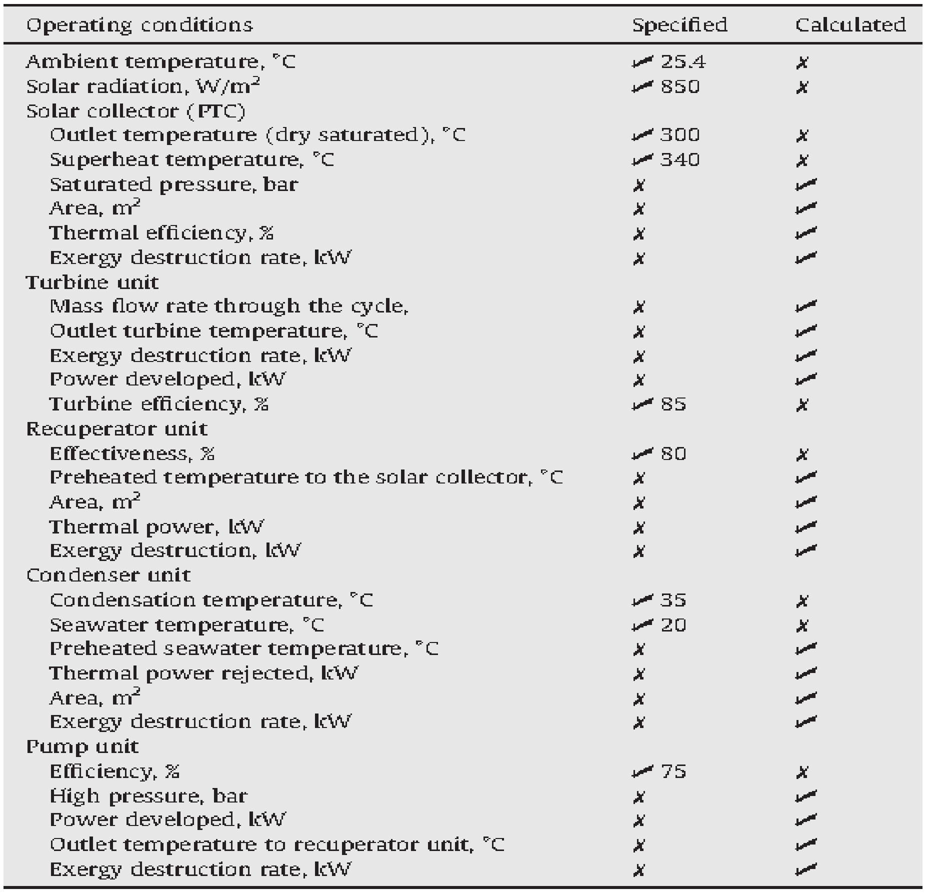
```
%Brine profile for rejection section:
Tbprej=min(Tbprec)-(Nrej.*Tstg);

%Stages dimensions and areas:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%The BPE for the stages:
BPErec=fbpe(Tbprec,Sr);

%Stage width:
Wstg=Mr./Vb;
Where Mr is the recycle mass flow rate, and Vb is the brine load (chamber load)
%the vapor allowable density kg/m3:
RAWv=1.299e-006*(Tbp.^3)-0.0001315*(Tbp.^2)+0.00728.*Tbp-0.1134;

%Stage length:
Lstg=min(Dsp)./(min(RAWv).*Vv.*Wstg);

%The brine density in the stages is obtained:
RAWbrec=fraw(Tbprec,Sr);

%Pressure drop profiles bar:
T1=TBT-Tstg;
Tbprec2=T1-(Tstg.*Nhr);
Pp=fp(Tbprec);
Pp2=fp(Tbprec2);

%Gate height:
GHrec=(Mr./(Cd.*Wstg)).*((2.*RAWbrec.*(Pp-Pp2).*1000).^(-0.5));

%Brine pool hieght:
Hrec=0.2+GHrec;

%The non-equlibrieum allowance NEA:
NEArec=fnea(TBT,Hrec,Vb);

%The temperature drop in demister is assumed negligible in comparison with the values of BPE
%and NEA. Therefore, the vapor temperature in the 1st stage is obtained:
Tvrec=Tbprec-BPErec-NEArec;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%The BPE for the stages:
BPErej=fbpe(Tb,Sb);

%The brine density in the stages is obtained:
RAWbrej=fraw(Tbp(max(Nt)-1),max(Srp));

%Pressure at stages n & n-1:
Pn=fp(Tb);
Pbrej=fp(Tbprej(max(Nrej)-1));

%Gate height:
GHrej=(Bp(max(Nt)-2)./(Cd.*Wstg)).*((2.*RAWbrej.*(Pbrej-Pn).*1000).^(-0.5));

%Brine pool height:
Hrej=0.2+GHrej;

%The non-equlibrieum allowance NEA:
NEArej=fnea(Tbprej(max(Nrej)-1),Hrej,Vb);

%The temperature drop in demister is assumed negligible in comparison with the values of BPE
%and NEA. Therefore, the vapor temperature in the 1st stage is obtained:
Tvrej=Tbprej-BPErej-NEArej;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Tvp=horzcat(Tvrec,Tvrej);
GH=horzcat(GHrec,GHrej,GHrej,GHrej);
H=horzcat(Hrec,Hrej,Hrej,Hrej);
%The vapor temperature in the 1st stage is used to calculate U and LMT:
```

```
LMTrec=(Tfp(2)-Tfp(1))./(log((Tvp(1)-Tfp(1))./(Tvp(1)-Tfp(2))));
%the overall heat loss for recovery stage:
U1=fu(Tvp(1));

%The area of the heat recovery stage Arec is then calculated:
Trm=(Tfp(1));
Arec=(Mr.*fcp(Trm,Sr).*(Tfp(1)-Tfp(2)))./(U1.*LMTrec);

%Tube length in the cooling water condenser:
Ltrec=fraw(Trprec,Sr).*((Dtihr.^2)./4).*Vf.*(Arec./(Dtohr.*Mr));
%Number of tubes in heat recovery:
Ntrec=Arec./(pi.*Dtohr.*Ltrec);

%%Heat rejection heat transfer area Arec:
LMTrej=((min(Tvp)-min(Tfp))-(min(Tvp)-Tsea))./(log(abs((min(Tvp)-min(Tfp))./(min(Tvp)-Tsea))));
Un=fu(min(Tvp));
Arej=(Mft.*fcp(Tsea,Sf).*((min(Tfp))-Tsea))./(Un.*LMTrej);

%Tube length in the cooling water condenser:
Ltrej=fraw(Tfprej,Sf).*((Dtir.^2)./4).*Vf.*(Arej./(Dtor.*Mft));

%Number of tubes in heat recovery:
Ntrej=Arej./(pi.*Dtor.*Ltrej);
```

4.7 Solar Powered Reverse Osmosis (RO-PEX) Model: Case Study

In this model, solar organic Rankine cycle with direct vapor generation technique is combined with reverse osmosis-pressure exchanger (RO-PEX) desalination plant. The system is designed to produce an amount of 145.8m^3/h of fresh water. The Design specifications of such plant are obtained based on Sharm El-Skeikh desalination plant [Desalination 261 (2010) 138–147].

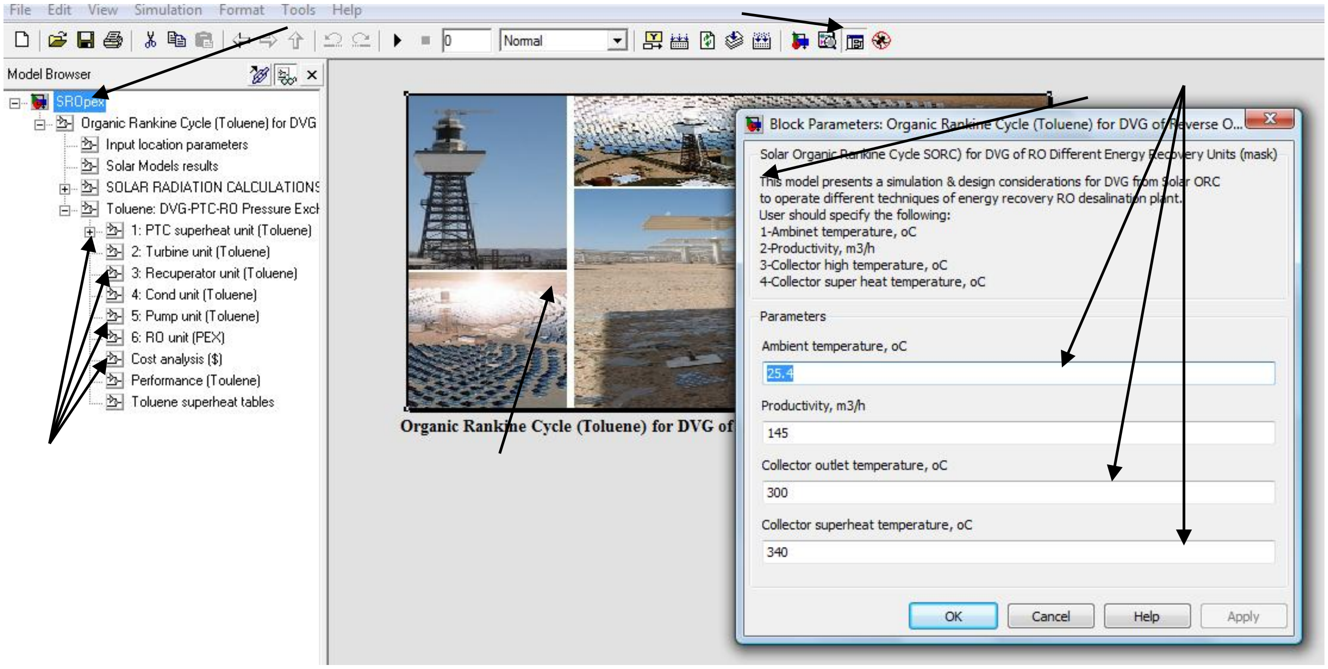
Specifications of solar ORC for RO desalination process.



Operating conditions	Specified	Calculated
Ambient temperature, °C	✓ 25.4	✗
Solar radiation, W/m^2	✓ 850	✗
Solar collector (PTC)		
Outlet temperature (dry saturated), °C	✓ 300	✗
Superheat temperature, °C	✓ 340	✗
Saturated pressure, bar	✗	✓
Area, m^2	✗	✓
Thermal efficiency, %	✗	✓
Exergy destruction rate, kW	✗	✓
Turbine unit		
Mass flow rate through the cycle,	✗	✓
Outlet turbine temperature, °C	✗	✓
Exergy destruction rate, kW	✗	✓
Power developed, kW	✗	✓
Turbine efficiency, %	✓ 85	✗
Recuperator unit		
Effectiveness, %	✓ 80	✗
Preheated temperature to the solar collector, °C	✗	✓
Area, m^2	✗	✓
Thermal power, kW	✗	✓
Exergy destruction, kW	✗	✓
Condenser unit		
Condensation temperature, °C	✓ 35	✗
Seawater temperature, °C	✓ 20	✗
Preheated seawater temperature, °C	✗	✓
Thermal power rejected, kW	✗	✓
Area, m^2	✗	✓
Exergy destruction rate, kW	✗	✓
Pump unit		
Efficiency, %	✓ 75	✗
High pressure, bar	✗	✓
Power developed, kW	✗	✓
Outlet temperature to recuperator unit, °C	✗	✓
Exergy destruction rate, kW	✗	✓

To enter the input data to the model, user has to follow the steps as illustrated in the figure below:

- ✓ Start the model “SROpex.mdl” using matlab software.
- ✓ When the model starts, user should click on the explorer icon if the model wasn’t explored.
- ✓ By clicking on the main block name and double clicking on the main block, user can easily changing or assigning the input data such as ambient temperature (25.4°C), productivity (145.8m^3/h), and operating conditions such as temperatures (300 & 340°C).
- ✓ The main block always has the same name of the file “SROpex.mdl”.
- ✓ To specify the solar radiation data, user can click on the sub-block and double clicking on the block that named by “input location parameter” to specify the location of operation.


File Edit View Simulation Format Tools Help
0
Normal
Model Browser
SROpex
Organic Rankine Cycle (Toluene) for DVG
Input location parameters
Solar Models results
SOLAR RADIATION CALCULATIONS
Toluene: DVG-PTC-RO Pressure Exch
1: PTC superheat unit (Toluene)
2: Turbine unit (Toluene)
3: Recuperator unit (Toluene)
4: Cond unit (Toluene)
5: Pump unit (Toluene)
6: RO unit (PEX)
Cost analysis ($)
Performance (Toulene)
Toluene superheat tables
Organic Rankine Cycle (Toluene) for DVG of

Block Parameters: Organic Rankine Cycle (Toluene) for DVG of Reverse O...

Solar Organic Rankine Cycle SORC) for DVG of RO Different Energy Recovery Units (mask)

This model presents a simulation & design considerations for DVG from Solar ORC to operate different techniques of energy recovery RO desalination plant.
User should specify the following:
1-Ambinet temperature, oC
2-Productivity, m3/h
3-Collector high temperature, oC
4-Collector super heat temperature, oC

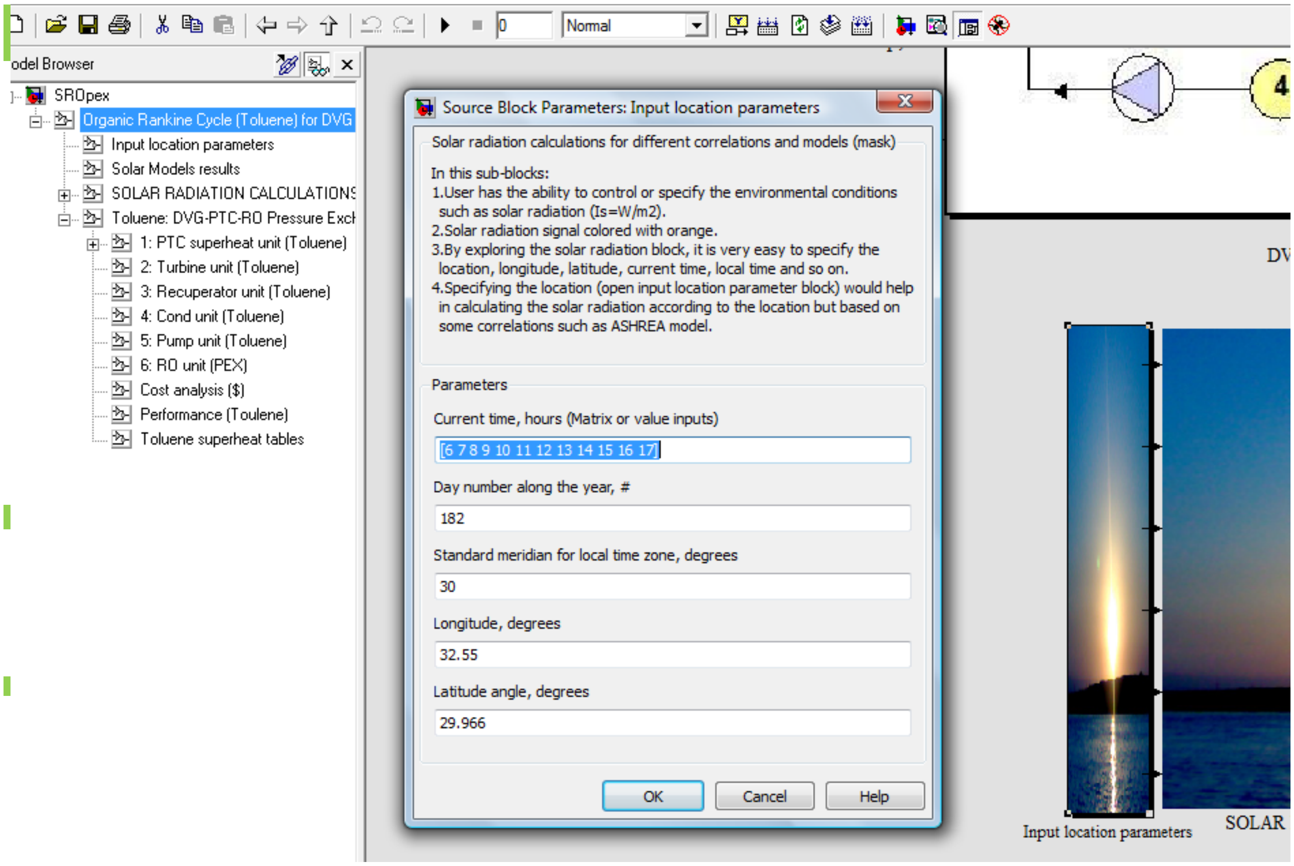
Parameters

Ambient temperature, oC: 25.4

Productivity, m3/h: 145

Collector outlet temperature, oC: 300

Collector superheat temperature, oC: 340


0
Normal
odel Browser
SROpex
Organic Rankine Cycle (Toluene) for DVG
Input location parameters
Solar Models results
SOLAR RADIATION CALCULATIONS
Toluene: DVG-PTC-RO Pressure Exch
1: PTC superheat unit (Toluene)
2: Turbine unit (Toluene)
3: Recuperator unit (Toluene)
4: Cond unit (Toluene)
5: Pump unit (Toluene)
6: RO unit (PEX)
Cost analysis ($)
Performance (Toulene)
Toluene superheat tables
4
DV
Input location parameters
SOLAR

Source Block Parameters: Input location parameters

Solar radiation calculations for different correlations and models (mask)

In this sub-blocks:
1.User has the ability to control or specify the environmental conditions such as solar radiation (Is=W/m2).
2.Solar radiation signal colored with orange.
3.By exploring the solar radiation block, it is very easy to specify the location, longitude, latitude, current time, local time and so on.
4.Specifying the location (open input location parameter block) would help in calculating the solar radiation according to the location but based on some correlations such as ASHREA model.

Parameters

Current time, hours (Matrix or value inputs): [6 7 8 9 10 11 12 13 14 15 16 17]

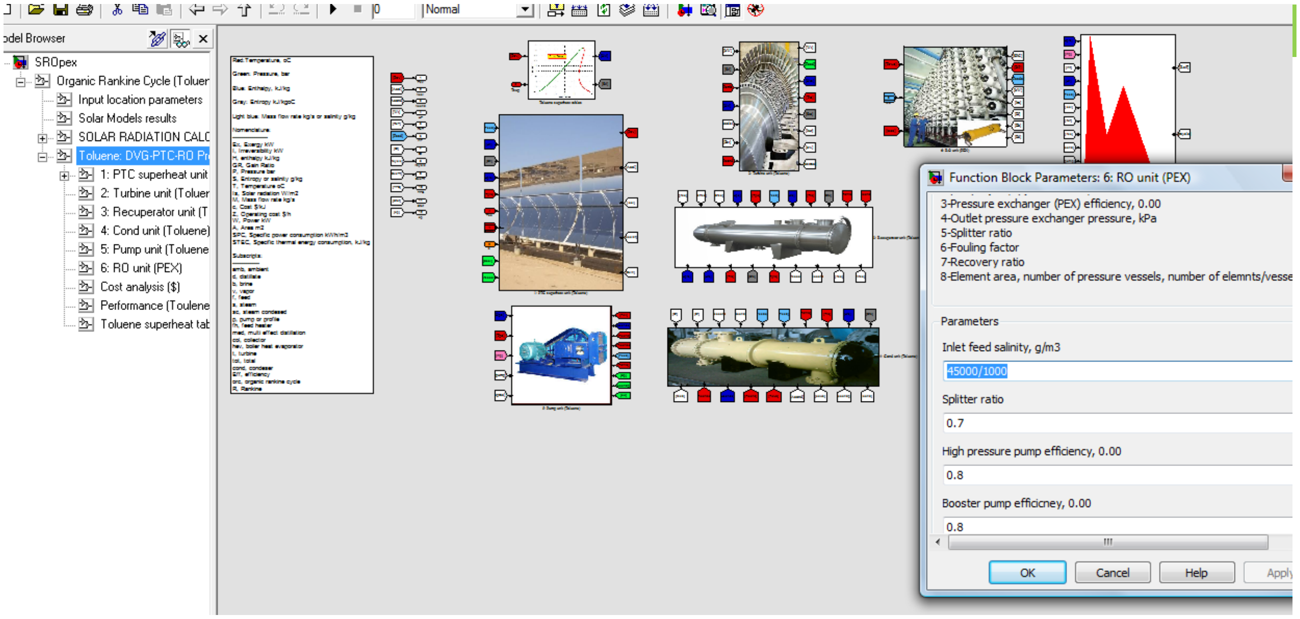
Day number along the year, #: 182

Standard meridian for local time zone, degrees: 30

Longitude, degrees: 32.55

Latitude angle, degrees: 29.966

User has to click on the main menu that contains the process units (for example: “Toluene: DVG-PTC-RO Pressure Exchanger recovery unit”). When the process units menu appears, user can easily double click on each unit individually to assign his own parameters. For example, user can assign the following for RO-PEX block:
1-Feed salinity, g/m3
2-pumps (HPP, Bp) efficiencies, 0.00
3-Pressure exchanger (PEX) efficiency, 0.00
4-Outlet pressure exchanger pressure, kPa
5-Splitter ratio
6-Fouling factor
7-Recovery ratio
8-Element area, number of pressure vessels, number of elements/vessel
User can easily get the results from the matlab desktop menu as shown in the figure below.



Function Block Parameters: 6: RO unit (PEX)

3-Pressure exchanger (PEX) efficiency, 0.00
4-Outlet pressure exchanger pressure, kPa
5-Splitter ratio
6-Fouling factor
7-Recovery ratio
8-Element area, number of pressure vessels, number of elemnts/vesse

Parameters

Inlet feed salinity, g/m3

45000/1000

Splitter ratio

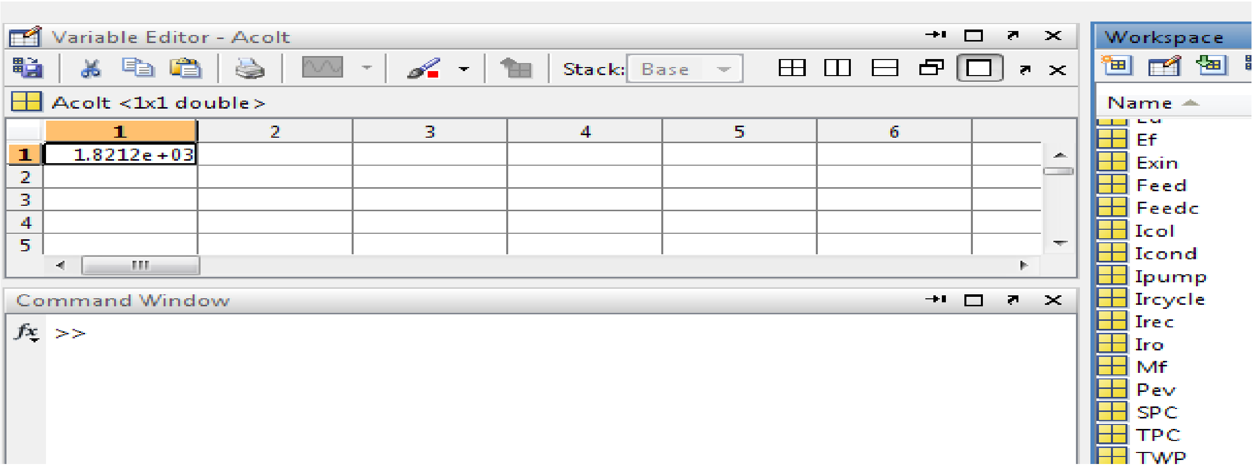
0.7

High pressure pump efficiency, 0.00

0.8

Booster pump efficicney, 0.00

0.8



Variable Editor - Acolt

Stack: Base

Acolt <1x1 double>

	1	2	3	4	5	6
1	1.8212e+03					
2						
3						
4						
5						

Command Window

```
>>
```

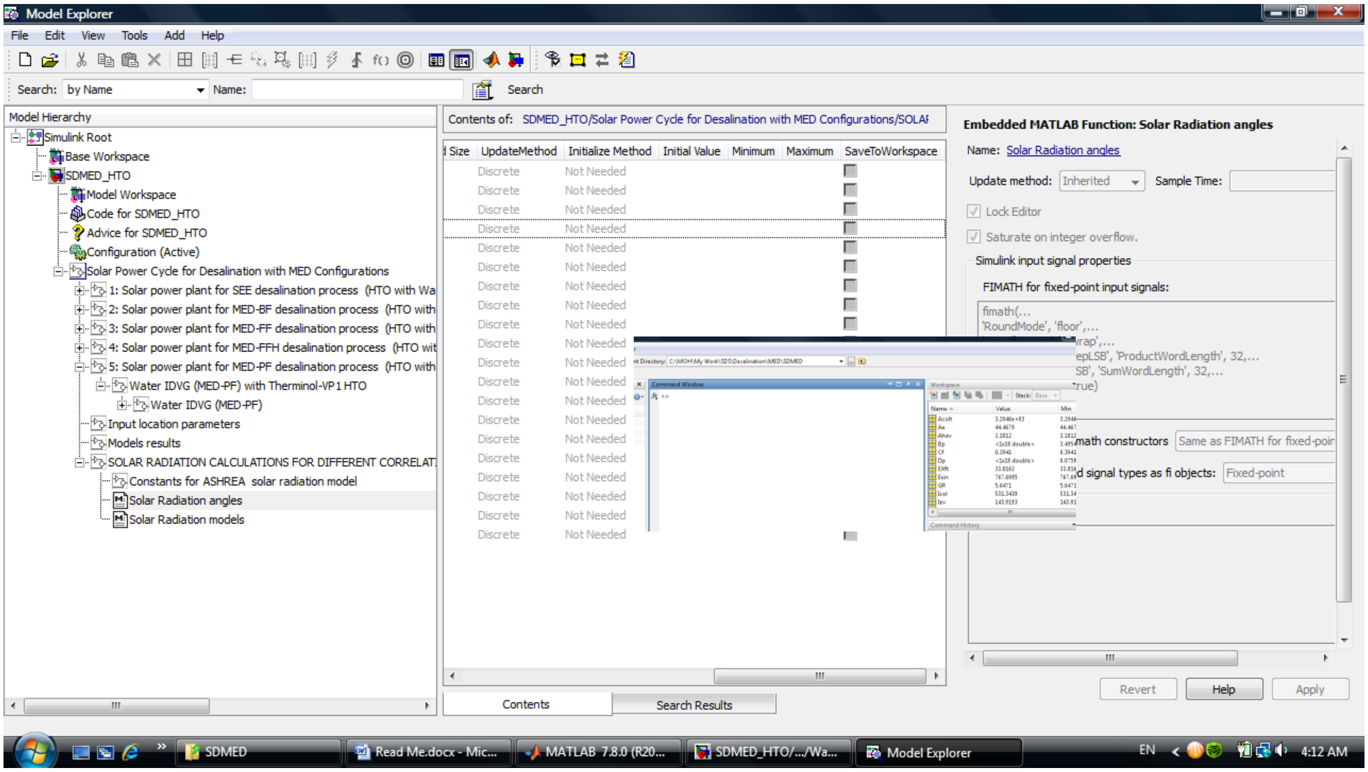
Workspace

Name
Ef
Exin
Feed
Feedc
Icol
Icond
Ipump
Ircycle
Irec
Iro
Mf
Pev
SPC
TPC
TWP

5. Model Run

How to run any model: Example

- First of all, Simulink models take the end path of *filename.mdl*.
- Suppose that user want to run the solar cycle for MED desalination plant. 1st run the matlab program. 2nd go to the current directory and specify the path that you put SDS folder. 3rd the model name (SDMED_HTO.mdl) will appear automatically in left menu of the current directory.
- Or try to write the model name (SDMED_HTO.mdl) in the work space after specifying the path of SDS main directory.
- Another easy way by double clicking on the model in its directory (SDMED_HTO.mdl) and the model will appear and run automatically without any need to a pre-run of matlab program: …\SDS\MED\SDMED\SDMED_HTO.mdl.
- When the simulation window of the proposed model appears, it is ease now to hold Ctrl+T to make the model run or go to "simulation" from tool bar and click on "start".
- To deal with any results or data by making some figures or storing the data; go to "launch model explorer" in tool bar menu and click on it. Model hierarchy will appear, then try to explore your model by clicking on (+) button. Click on the "user defined block" because this will help to clear out the block inputs and outputs. Then it is easy to mark the small square under the title of "save to workspace". Then run the model "Ctrl+T", the parameter that you select will appear in MatLab workspace.


Model Explorer
File Edit View Tools Add Help
Search: by Name Name: Search
Model Hierarchy
Simulink Root
Base Workspace
SDMED_HTO
Model Workspace
Code for SDMED_HTO
Advice for SDMED_HTO
Configuration (Active)
Solar Power Cycle for Desalination with MED Configurations
1: Solar power plant for SEE desalination process (HTO with Wa
2: Solar power plant for MED-BF desalination process (HTO with
3: Solar power plant for MED-FF desalination process (HTO with
4: Solar power plant for MED-FFH desalination process (HTO wit
5: Solar power plant for MED-PF desalination process (HTO with
Water IDVG (MED-PF) with Therminol-VP1 HTO
Water IDVG (MED-PF)
Input location parameters
Models results
SOLAR RADIATION CALCULATIONS FOR DIFFERENT CORRELAT
Constants for ASHREA solar radiation model
Solar Radiation angles
Solar Radiation models
Contents Search Results
Revert Help Apply
SDMED Read Me.docx - Mic... MATLAB 7.8.0 (R20... SDMED_HTO/.../Wa... Model Explorer EN 4:12 AM

Contents of: SDMED_HTO/Solar Power Cycle for Desalination with MED Configurations/SOLAI

Size	UpdateMethod	Initialize Method	Initial Value	Minimum	Maximum	SaveToWorkspace
	Discrete	Not Needed				☐
	Discrete	Not Needed				☐
	Discrete	Not Needed				☐
	Discrete	Not Needed				☐
	Discrete	Not Needed				☐
	Discrete	Not Needed				☐
	Discrete	Not Needed				☐
	Discrete	Not Needed				☐
	Discrete	Not Needed				☐
	Discrete	Not Needed				
	Discrete	Not Needed				
	Discrete	Not Needed				
	Discrete	Not Needed				
	Discrete	Not Needed				
	Discrete	Not Needed				
	Discrete	Not Needed				
	Discrete	Not Needed				
	Discrete	Not Needed				
	Discrete	Not Needed				
	Discrete	Not Needed				☐

Embedded MATLAB Function: Solar Radiation angles

Name: Solar Radiation angles

Update method: Inherited Sample Time:

☑ Lock Editor

☑ Saturate on integer overflow.

Simulink input signal properties

FIMATH for fixed-point input signals:

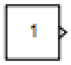
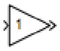
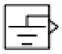
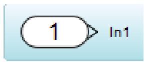
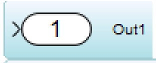
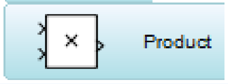
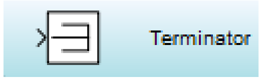
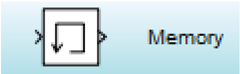
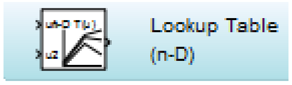
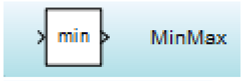
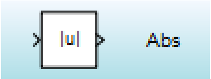
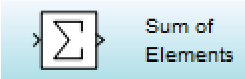
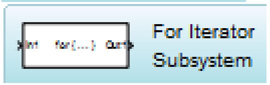
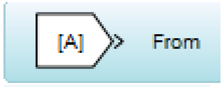
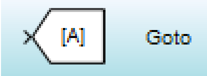
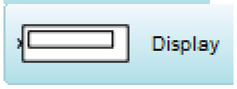
```
fimath(...
'RoundMode', 'floor',...
rap',...
epLSB', 'ProductWordLength', 32,...
SB', 'SumWordLength', 32,...
true)
```

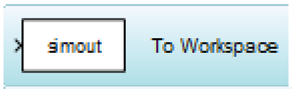
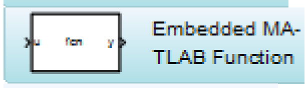
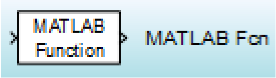
math constructors: Same as FIMATH for fixed-poir

d signal types as fi objects: Fixed-point

6. SimuLink Used Blocks

The following table contains all the blocks used to simulate and design solar desalination units.

Block name:	Shape:	Description:
Constant block		Output the constant specified by the 'Constant value' parameter.
Gain block		Element-wise gain (y = K.*u) or matrix gain (y = K*u or y = u*K).
Ground		Used to "ground" input signals. (Prevents warnings about unconnected input ports.) Outputs zero.
In block		Provide an input port for a subsystem or model.
Out block		Provide an output port for a subsystem or model.
Product block		Multiply or divide inputs.
Terminator		Used to "terminate" output signals. (Prevents warnings about unconnected output ports.)
Memory block		Apply a one integration step delay. The output is the previous input value.
Lookup table		Perform n-dimensional interpolated table lookup including index searches. The table is a sampled representation of a function in N variables. Breakpoint sets relate the input values to positions in the table.
Max/Min block		Output min or max of input. For a single input, operators are applied across the input vector.
Absolute block		y = |u|
Sum of elements		Add or subtract inputs.
For iterator block		A subsystem block template containing a for iterator, in-port and out-port block.
From block		Receive signals from the Goto block with the specified tag.
Goto block		Send signals to From blocks that have the specified tag.
Display block		Numeric display of input values.

To work space		Write input to specified array or structure in MATLAB's main workspace.
Embedded block		Embedded MATLAB block.
MatLab function		Pass the input values to a MATLAB function for evaluation.

Contacts

- For any further information, please contact SDS developers at the following links:
  - msherafeldin@gmail.com
  - mohammed.eldeen@suezuniv.edu.eg
  - +60 1139795943 (mob)
  - Engineering science department, faculty of petroleum & mining engineering, Suez Canal University, Suez, Egypt.
  - PV21 condo, Setapak, Kuala Lumpur Malaysia, 53300
  - Contact persons: Dr. **Mohamed Abdel Wahab Sharaf**.