ISSN 0280–5316
ISRN LUTFD2/TFRT--5543--SE

The Dynamic Simulation of a Humid Air Gas Turbine Cycle

Behroz Nanevazadeh

Department of Automatic Control
Lund Institute of Technology
November 1995

Department of Automatic Control
Lund Institute of Technology
P.O. Box 118
S-221 00 Lund Sweden

Document name MASTER THESIS	
Date of issue November 1995	
Document Number ISRN LUTFD2/TFRT--5543--SE	
Author(s) Behroz Nanevazadeh	*Supervisor* Bernt Nilsson, Rolf Johansson
	Sponsoring organisation

Title and subtitle

The Dynamic Simulation of a Humid Air Gas Turbine Unit

Abstract

The Humid Air Gas Turbine Cycle, HAT cycle, was modeled and simulated so that the dynamics of the system was studied. In the HAT cycle, dynamic models of a humidification tower, a recuperator, an air-cooled expander, and a water circulation system were made, using models of fluid compression, expansion, mass and heat transfer, mass and energy balances, and heat generation.

An analysis of the non-linearities of the discretecized humidification tower resulted in a proposed stabilization of the tower via a state-feedback control law. Thereby, the transient thermal characteristics of the discretecized tower was controlled by pole-placement.

The simulated results were compared to operational data from different power plants by thermodynamic performance criteria such as energy utilisation factor, fuel energy saving ratio, and artificial thermal efficiency. The HAT cycle model was implemented in the object-oriented modeling language, Omola, and simulated in the OmSim environment , which are both created at The Department of Automatic Control, Lund Institute of Technology.

Key words
Gas Turbine Unit, Humidification Tower, Humidity, Air-Cooled Expander, Mass and Heat Transfer, Mass and Energy Balance, Recuperator, Coefficients of Performance, Water Circulation System, Air-Water System.

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title 0280–5316		*ISBN*
Language English	*Number of pages* 76	*Recipient's notes*
Security classification		

The report may be ordered from the Department of Automatic Control or borrowed through the University Library 2, Box 1010, S-221 03 Lund, Sweden, Fax +46 46 110019, Telex: 33248 lubbis lund.

'So God created man in his own image'

Genesis 1:27

A *created* model is not perfect,
you know.
It's mostly the low order non-linear models
that capture the essential characteristics anyway!

A control engineers advice to a
novice, deep in thoughts. LTH 1995.

Contents

List of Figures

Chapter 1

Introduction

The purpose of this report is to present **The Dynamic Simulation of The Humid Air Gas Turbine Cycle, HAT cycle** . The **Hat cycle** is a modified gas turbine unit which can reach high coefficients of performance. Dynamic models provide information and insight about the size, arrangement of the units, their interaction, and stability among many other objectives. The uncertainty of process parameters such as the two phase counter-current interaction of the air-water system in the humidification tower of the HAT cycles, the heat transfer constants, the water flow rate and the critical placement of the recuporator, are concepts that an engineer can successfully challenge with the insight from *acceptable* dynamic models. The off-line study of the *start-up* and *shut-down* operating conditions by simulating dynamic models are also of significant value to the power engineer.

An understanding of the process fundamentals is required to be able to start on the path of modeling a power plant. A power plant such as the HAT cycle is a complex process, therefore the structural hierarchy of the process should be studied and analysed. By studying these details it is possible to find a *guideline* for the decomposition of the process into units and subunits. Dynamic models of these units or subunits show their time-dependent behaviour, and thereby the transient behaviour of the power plant, if the models are put together as building blocks of the process. At all the different stages the basic tools of modeling which consist of mass, energy, and momentum balances were applied in their time-varying forms.

The work is divided into four areas of discussion. These four areas are

- Process Description
- Process Unit Models and Control Systems
- Comparative Thermodynamics of Power Plants
- Validation by Simulation Results

The units are described mathematically and encapsulated in model units which interact with each other dynamically through contact surfaces here named **terminals**. So, the power plant unit states can be observed as a function of different inputs and time in an input-output model.

By the use of performance data from operational plants a comparison is made to validate the high performance data from that of the modeled Humid Air Gas Turbine Unit. Therefore, different power plants are introduced, and represented by thermodynamic drift data. Another purpose of introducing these criteria is to find parameters that could be to simulate modestly valid and representable simulation results.

The simulation results of the following systems are presented:

- The Humidification Tower
- The Waste Heat Recuporator
- The Tank System
- The Thermodynamic Performance of the HAT cycle

The tank system keeps the volume of the circulating water poring into the tank constant by a discrete-time PI controller. The air cooled expander is a proposed model for the mixing of a *bypass* flow out of the humidification tower with that of the exhaust gas leaving the combustion chamber. The Waste Heat Recuporator, WHR, placed at the outlet of the expander unit in the HAT cycle gives rise to a non-linear system which is studied by a phase-plane analysis for the purpose of understanding its behaviour.

The non-linearities in the humidification tower model, due to the effects of the simultaneous mass and heat transfer, need to be regarded if the flows in the cycle are manipulated by the process control systems. The humidification tower is therefore stabilized via a state feedback control law.

Finally, the thermodynamic performance of the HAT cycle is also shown by simulation runs. This is an attempt to verify the models. Parameters are adjusted to describe the 'real-world' situation. Similar high performance plants provide data which HAT cycle should attain. This is also an attempt to design an experiment to verify the overall power plant model.

Chapter 2

Process Description

2.1 Basic Gas Turbine Unit

The most basic gas turbine unit is shown in the figure 2.1.

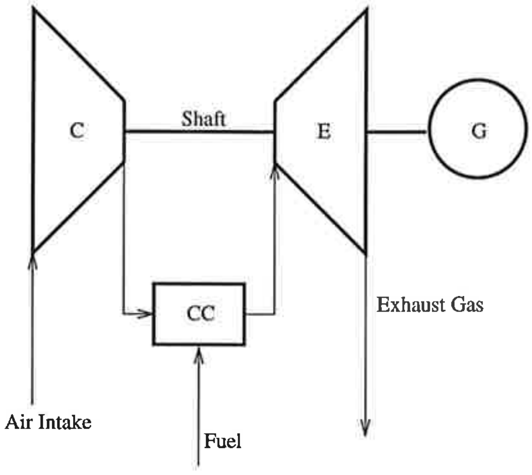

Figure 2.1: A Gas Turbine Unit

This unit is considered as an open cycle. The air is taken in by a rotary compressor, **C**, sitting on a common shaft with the expander, **E**. The gas turbine unit also consists of a combustion chamber, **CC** and generator, **G**. So, the basic gas turbine unit consists of the following 4 subunits:

- Expander
- Compressor
- Combustion Chamber
- Generator

An attempt on modeling the basic gas turbine unit should include a library of the above mentioned subunits [1, 6].

2.2 Modification of the Basic Unit

A modification of the basic cycle in order to achieve higher thermodynamic efficiency would result in the addition of other subunits into the process [9, 11]. Figure 2.2 shows the Humid Air Gas Turbine Cycle, *HAT cycle.* The basic gas turbine unit is modified to include a waste heat recuporator, **Rec**, intercooling, **IC**, between the low compressor, **LC**, and the high compressor, **HC**. The circulation water is controlled by a water tank system, **Tank**. The expander unit, **E**, is fitted with a *bypass* flow for the purpose of a cooling the expander blades. The air cooling system consists of a split unit, **S**, where a certain precentage of the flow at the outlet of the humidification tower is redirected into the expander unit.

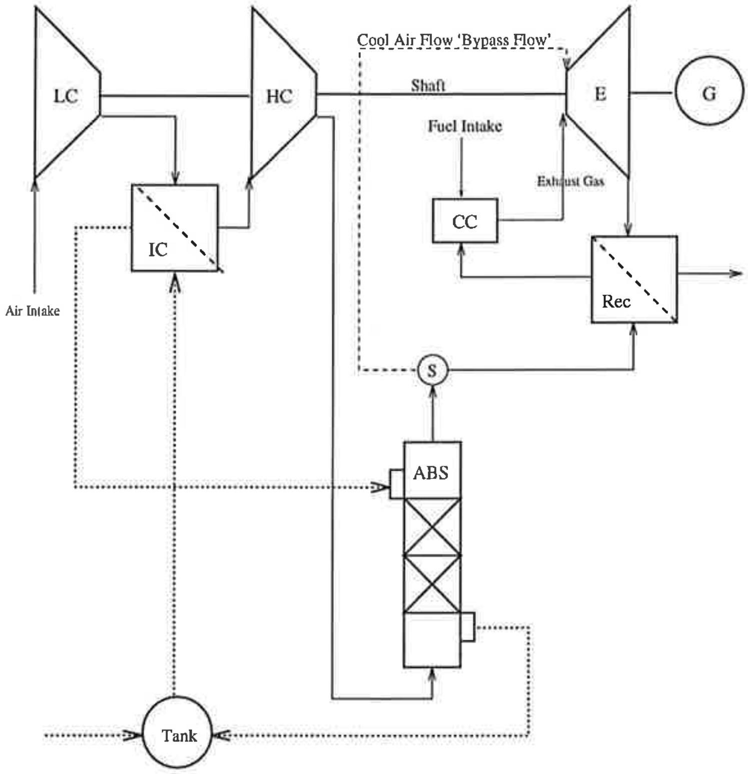

Figure 2.2: The Humid Air Gas Turbine Cycle

The gas turbine unit consists of two compressors, **LC** and **HC**, sitting on the same shaft. The **LC** takes in air compresses it to a set pressure ratio. This air then enters the **IC**, connected at the outlet of the compressor. The air is then cooled down in the intercooler before it enters the high compressor, **HC**. This is therefore a one stage intercooling process between the low compressor, **LC**, and the high compressor, **HC**. The high compressor then compresses the air to a final value before it enters the humidification tower placed at its outlet.

Both the intercooler, **IC**, and the waste heat recuporator, **Rec**, are heat exchangers. Heat exchangers are connected to provide heat transport between flowing fluids. A recuperator is placed at the outlet of the expander, **E**, to exchange heat with the air entering the combustion chamber, **CC**. This design causes useful heat otherwise lost to be recuperated into the system. The **IC** unit exchanges heat from the hot gas leaving the **LC** with the circulation water. The heat won back at the **LC** is thereby kept in the circulating water system. The circulating water then enters the humidification tower, **ABS**.

There is a need to cool down the air leaving the **HC** before it enters the recuporator, **Rec**, so that a maximum heat exchange can occur with the air leaving the expander unit. This required cooling down of the streaming air is carried out with an Humidification Tower, **ABS**. This unit is called an **ABS** since it is an attempt to model the absorbtion of humidity into the air system. The air traveling up through the tower meets the circulating water traveling down, in a counter-current flow system. The air is cooled down by meeting the water in stages and transferring heat to the water side. This transfer of heat elevates the energy of the water on the contact surface so that water is evaporated into the air side by a mass transfer mechanism. Therefore, a simultaneous mass and heat transfer takes place between the air and water systems meeting in a counter-current flow. The water evaporated increases the humidity of the air in the tower. This rise in the mass flow in itself causes an increase in the total efficiency of the plant. Hence, the Humid Air Gas Turbine Cycle is classified as a Evaporative Gas Turbine Cycle [5].

There is a need to run existing plants with different fuels of different energy values. The possibly higher energy value of the new fuel causes a rise in temperature at the expander inlet. This phenomena is compensated by a proposed model of an air cooling system. The air cooling system reacts by *bleeding* the compressed air in the power plant and sending a *bypass* flow into the expander. A split unit placed at the outlet of the Humidification tower can redirect a certain procent of the cool air leaving the tower to pass it through the system and inject it directly into the expander.

The circulating water is pored into a tank which regulates the mass flow of the circulating water system and keeps the volume of water in the tank constant. The tank system, **Tank**, should compensate for the water being evaporated in the humidification tower and the pressure loss on the water side leaving the humidification tower.

Chapter 3

Process Unit Models

3.1 The Gas Turbine Unit

A basic gas turbine unit was shown in figure 2.1. The compressor, which sits on the same shaft as the turbine, takes in air which is then compressed to a operating pressure. The air enters the combustion chamber where fuel is injected and combustion occurs. The result is the expansion of high pressure hot exhaust gas through the expander. The expansion of exhaust gas in the expander results in a shaft work, that in turn rotates the compressor blades. The excess energy, the difference between expander and compressor work, is used to generate electricity in the generator.

Although dynamic models exist for the basic gas turbine unit [20, 22, 23], a static model is used to present the expander and compressor units. This will further ease the study of the dynamics of the humidification tower which is unique for these types of high performance power plants.

3.1.1 A Model of the Expansion and Compression of an Ideal Gas

The assumption that the expansion of the exhaust gas through the expander is an isentropic process and that the exhaust gas is a compressible fluid results in a static model, which is used in the modeling of the expander unit, **E**, in the Hat cycle. The relation between the pressure and volume for an isentropic expansion or compression of an ideal gas is

$$PV^{\gamma} = constant \tag{3.1}$$

where $\gamma = \frac{C_p}{C_v}$. C_p is the heat capacity of the gas at constant pressure and C_v is the heat capacity of the gas at constant volume. This relation holds approximately true, since γ has been taken as a constant. Although, ***Hussain and Seifi*** [23] have studied the variation of this important process parameter on their model.

The work done by the expansion of the exhaust gas is given by

$$dW = d(PV) = dp \cdot V + P \cdot dV \tag{3.2}$$

Equation 3.2 relates the produced work to the change in pressure and volume of the compressible expanding fluid, in this case the exhaust gas. Then, it is assumed that $dV \approx 0$ which implies that a thermodynamic control volume is being considered. By evaluating the equation

$$W = \int_{P_1}^{P_2} V dp \tag{3.3}$$

for an ideal gas in an isentropic process, a relation for work as a function of pressure ratio over the turbine is obtained. The relation for work in expanding or compressing an ideal gas isentropically at a constant volume is

$$W = \frac{\gamma}{\gamma - 1} P_1 V_1 [1 - (\frac{P_1}{P_2})^{\frac{\gamma}{\gamma - 1}}] \tag{3.4}$$

Equation 3.4 is the result of combining the ideal gas equation $PV = nRT$, and equation 3.3 resulting in the work needed in compressing an ideal gas from P_1 to P_2. $P_2 > P_1$ this would result in a compression and $W > 0$.

$$W = \begin{cases} W < 0 & : \text{expansion over the expander,} \\ W > 0 & : \text{compression over the compressor} \end{cases} \tag{3.5}$$

This static model for the expansion or compression of an ideal gas can be used to model the expander and compressor in the power plant unit shown in figure 2.2.

3.1.2 The Air Cooled Expander Model

When the temperature of the exhaust gas leaving the combustion chamber exceeds the expander blades operating temperature of the blades on the expander, cool air is bled from a certain point in the process and injected in the expander unit. This allows for the mixing of the hot exhaust gas with the cool air. Complicated models exist [5]. A model is proposed for the mixing of the hot exhaust gas entering the expander and the *bypass flow*. The mixing model for these two flows is assumed to occur in a control volume at the expander inlet, so that the dynamics of the mass flow and energy flow can be considered.

Figure 3.1 shows a simple block diagram over the separation of the Expander into a mix unit and and an expansion unit. The exhaust gas leaving the combustion chamber is mixed with cool air flowing from the split unit i.e the *bypass* flow. This takes place in the mix unit which models the inlet to the gas turbine. The two fluids mix idealy and the resulting fluid enters the expansion phase in the gas turbine unit. The continuity equations gives:

$$\frac{dm}{dt} = w_{out}^{CC} + w_{in}^{CoolAir} - w_{out}^{mix} \tag{3.6}$$

where w stands for mass flow with the units $(\frac{kg}{s})$. It should be pointed out that

$$w_{out}^{mix} = w_{in}^{E}$$

there the outflow from the mix unit is considered as the inflow into the expansion unit. Equation 3.6 states that the mass flow of the exhaust gas at the gas turbine inlet plus the mass flow of the cool air flow entering the gas turbine, subtracted by the mass flow of exhaust gas entering the expansion phase, is equal to the mass of gas accumulated in the control volume as a function of time. It is also assumed that an ideal mix is taking place. This is important since the energy balance is given by

$$\frac{d(mH_{out}^{mix})}{dt} = w_{in}^{CoolAir} \cdot H_{in}^{CoolAir} + w_{out}^{CC} \cdot H_{out}^{CC} - w_{out}^{mix} \cdot H_{out}^{mix} \tag{3.7}$$

Using the equation

$$\frac{d(mH_{out}^{mix})}{dt} = \frac{dm}{dt} H_{out}^{mix} + \frac{dH_{out}^{mix}}{dt} m \tag{3.8}$$

the following ordinary differential equation can be obtained to describe the thermal dynamics of the exhaust gas enthalpy as a function of the mix between the enthalpy of the cool-air from the split unit and exhaust gas from the combustion chamber.

$$m \cdot \frac{dH_{out}^{mix}}{dt} = w_{out}^{CC} \cdot (H_{out}^{CC} - H_{out}^{mix}) + w_{in}^{CoolAir} \cdot (H_{in}^{CoolAir} - H_{out}^{mix}) \tag{3.9}$$

It is assumed that the two fluids mix perfectly and that PV-work is negligible so that the internal energy of the system can be taken as equal to the enthalpy $\Delta U \approx \Delta H$.

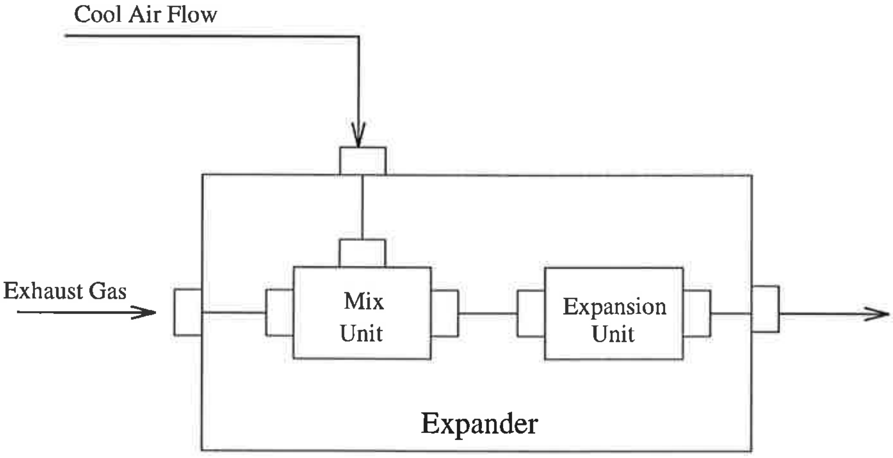

Figure 3.1: The Mix and The Expander Units

The Omola code for this construction is:

```
TurbineModel      ISA Model WITH

Parameters:
GasCnst   ISA Parameter WITH default:=8.314;END;
MolMass   ISA Parameter WITH default:=28;END;
```

```
Xi         ISA Parameter WITH default:=1.4;END;
Cpg        ISA Parameter WITH default:=1.4;END;
m          ISA Parameter WITH default:=5;END;

Terminals:
UnitConIn   ISA InCon;
UnitConOut  ISA OutCon;
CoolAirFlow ISA SimpleInput;
CoolAirEnth ISA SimpleInput;

Variables:
ra,Work,k     ISA Variable;
Del1,Del2     ISA Variable;
EnthTurbine   ISA Variable WITH initial:=1000;END;
FlowTurbine   ISA Variable WITH initial:=10;END;

Equations:
ra                := UnitConOut.Pout/UnitConIn.Pin;
k                 :=  Xi/(1-Xi);
UnitConOut.Wout :=FlowTurbine;
Work  := k*FlowTurbine*GasCnst*EnthTurbine/(Molmass*Cpg)*(ra^(-1/k)-1);
UnitConOut.Hout := EnthTurbine*ra^(-1/k);
FlowTurbine       :=   UnitConIn.Win + CoolAirFlow;
m*EnthTurbine'     = -FlowTurbine*EnthTurbine+Del1+Del2;
Del1:=CoolAirFlow*CoolAirEnth;
Del2:=UnitConIn.Win*UnitConIn.Hin;

END;
```

This is implementation in the Object Oriented Modeling Language, **Omola** [15, 16, 17]. The **TurbineModel** has **terminals** whereby the model can react with the environment or other models. These terminals are:

- UnitConIn
- UnitConOut
- CoolAirFlow
- CoolAirEnth

The terminals UnitConIn and UnitConOut are of the same data type as InCon and OutCon which appear as Super-Class in the connections between the models, in this program. The **TurbineModel** has two other terminals which are the CoolAirFlow and CoolAirEnth. These are declared as SimpleInput. This means that the turbine can only take in cool air with a cool air enthalpy.

```
SplitModel ISA Model WITH
% The bypass flow s controlled by the split unit
% which redirects the flow.
```

```
Parameters:
X           ISA Parameter WITH initial:=0.05;END;

Terminals:
InFlow      ISA InCon;
OutFlow     ISA OutCon;
CoolAirFlow ISA SimpleOutput;
CoolAirEnth ISA SimpleOutput;

CoolAirFlow  :=X*InFlow.Win;
CoolAirEnth  :=InFlow.Hin;
OutFlow.Hout :=InFlow.Hin;
OutFlow.Wout :=(1-X)*InFlow.Win;
OutFlow.Pout :=InFlow.Pin;

END;
```

If the enthalpy of the exhaust gas leaving the combustion chamber **CC** and the enthalpy of the exhaust gas entering the expansion phase after the mix unit, are greater than the conventional temperature limit then the parameter **X** in the Split unit is increased accordingly. The incremental rise in the value of **X** models the percent of flow through a valve. This rising need for an increasing *bypass* flow rate is thereby translated over to the Split unit, **S**.

3.1.3 The Compressor System and Model

Figure 3.2 shows the compressor system in the Hat Cycle. The **LC** takes in air which is then compressed to a prescribed value P_i, where **i** stands for intermediate. The compressed air is then cooled in the **IC** before it enters the **HC**. As proven and briefly discussed earlier the work required to compress air isentropically is given by

$$W = \frac{\gamma}{\gamma - 1} P_1 V_1 [1 - (\frac{P_1}{P_2})^{\frac{\gamma}{\gamma-1}}]$$

A decrease in the work output of the compressor system by intercooling results in the increased work output of the gas turbine. A study of the intercooler is done in the section **Heat Exchanger Models**, where dynamic and static models of the heat exchangers are discussed.

It is proven theoretically that the minimum compression work is given by the pressure relation

$$P_i^2 = P_1 \cdot P_2 \tag{3.10}$$

where P_i is the *intermediate* pressure of the air leaving the outlet of the low compressor [21]. This is achieved by cooling the air leaving the intercooler to the same level as the air entering the low compressor [14]. However, the sound decision to allow the compressor to utilize the same work input in order to increase the evaporative capacity of a relatively warmer air entering the humidification tower is an interesting option which is outside the scope of this thesis.

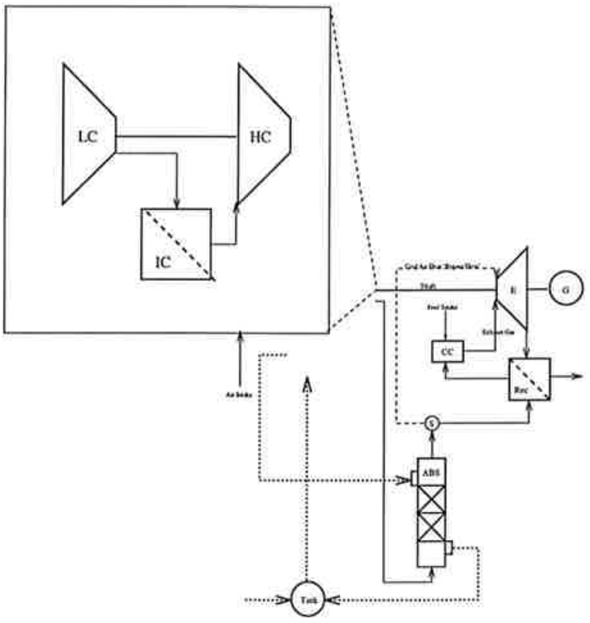

Figure 3.2: The Compressor System Model

3.2 The Humidification Tower Model

Figure 3.3 shows the humidification tower, **Abs**, and the tank model, **Tank**. The tower is modeled as an absorption towers used in the chemical industry. These industrial towers are mainly used in the separation processes [19]. However, the purpose of the tower in the Humid Air Gas Turbine Cycle is to cool the air streaming out of the high compressor, **HC**, by bringing the air into contact with cool circulating water. The water comes into contact with the air in a counter current process. The transfer of heat from the air side to the water side causes the evaporation of water into the air, whereby the humidity of the air flowing in the system is increased which has direct effect on the performance of the system. Work produced by the expansion of exhaust gas in the expander is, among other parameters, proportional to this increase in the mass flow.

The cooling of air is carried out by the transfer of heat from the hot air to the water whereby water is evaporated from the surface of the contact area between the two fluids. This is an example of simultaneous mass and heat transfer between contacting fluids [18]. This causes the humidity of the air to increase. Thereby, the model of a humidification tower should encapsulate the process of simultaneous mass and heat transfer between counter current flow of hot air and cool water.

A tank model, **Tank**, is modeled in the program to *catch* the flow of the circulating water. The tank is controlled to keep a constant volume of water to compensate for the loss of water due to the humidification of the air system.

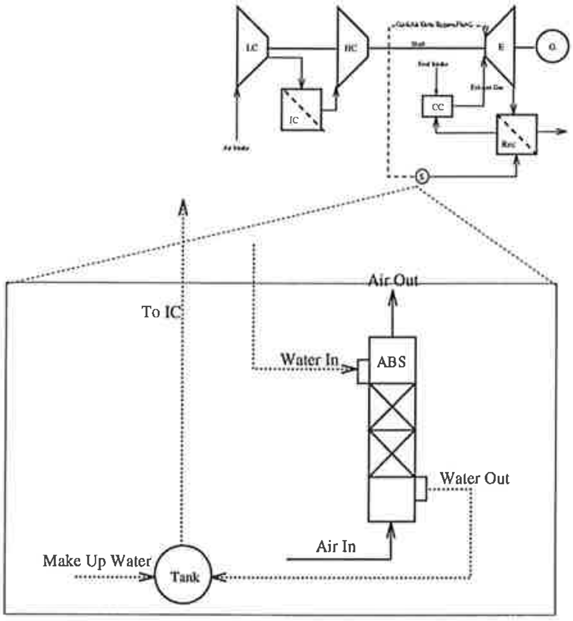

Figure 3.3: The Humidification Tower in the Hat Cycle

3.2.1 The Air Water System Model

The presence of two phases creates considerable complications in the system description and problem formulation. Thereby, many assumptions are made to simplify the modeling process. According to *Coulson and Richarson*, [25] on the topic of multiphase flow systems:

> The complexity of the flow is so great that design methods depend very much on an analysis of the behaviour of such systems in practice and, only to a limited extent, on theoretical predictions.

An assumed first step in the modeling of the tower system would be to *cut* across the system as shown in figure 3.4. Then it would be favourable to further decompose the *cut* into the subsystems:

- Air System
- Water System
- Film System

By carrying out this decomposition into the above named subsystems it will become possible to model each subsystem separately [7]and then to define the interaction between these submodels.

The Air System Model

The Air System is shown in figure 3.5. The air system is considered as a control volume. It is assumed that in this model the kinetic energy and potential

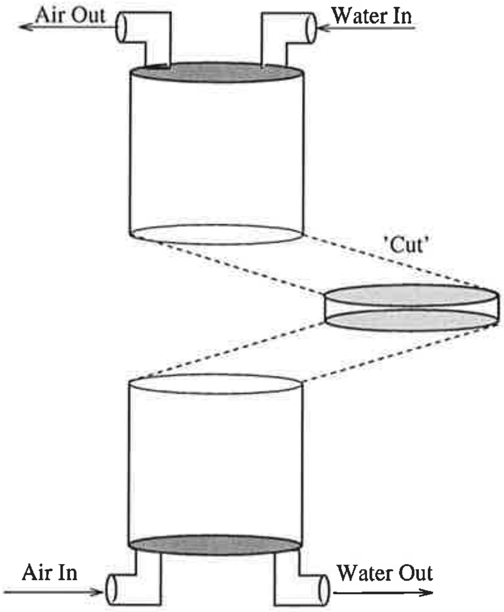

Figure 3.4: A *Cut* into the Humidification Tower

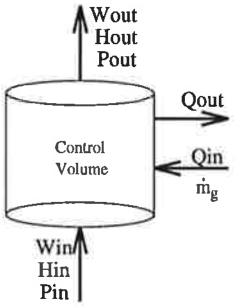

Figure 3.5: A Diagram over the Air System

energy of the stationary control volume is negligible due to very high operational pressure. The high pressure implies that the volume has high energy per unit volume. There is of course no shaft work and the dynamic pressure due to inflow and outflow in the control volume has been also taken as negligible compared to the high operation pressure in the tower. Therefore, the internal energy, **U**, of the system can be taken as equal to the enthalpy, **H**, $\Delta H \approx \Delta U$. Now, the continuity equations are applied. The mass balance of the system gives

$$\frac{dm^{as}}{dt} = w^{as}_{in} - w^{as}_{out} + \dot{m}_g \tag{3.11}$$

The flow into the control volume, w^{as}_{in}, there **as** stands for the Air System, plus the amount of water evaporated per second $\dot{m}_g$, subtracted by the mass flow of air flowing out of the control volume, w^{as}_{out}, is equal to the water accumulated in the control volume as a function of time. The energy balance over the system gives

$$\frac{d(m^{as} \cdot H^{as}_{out})}{dt} = w^{as}_{in} \cdot H^{as}_{in} - w^{as}_{out} \cdot H^{as}_{out} + Q^{as}_{in} - Q^{as}_{out} \tag{3.12}$$

Q_{out} is the heat transfer from the control volume to the contact surface with the water side. Q_{in} is the heat transferred back to the air side by the evaporated water. The relation

$$\frac{d(m^{as} \cdot H_{out})}{dt} = \frac{dm^{as}}{dt} \cdot H^{as}_{out} + \frac{dH^{as}_{out}}{dt} \cdot m^{as} \tag{3.13}$$

gives the ODE for the energy balance over the air side of the *cut*

$$m \cdot \frac{dH^{as}_{out}}{dt} = w^{as}_{in} \cdot (H^{as}_{in} - H^{as}_{out}) - \dot{m}_g \cdot H^{as}_{out} + Q^{as}_{in} - Q^{as}_{out} \tag{3.14}$$

H^{as}_{in} is the enthalpy of the air flowing into the control volume, and H^{as}_{out} is the enthalpy of the air flowing out of the control volume.

The Water System Model

The Water System is shown in figure 3.6. The same assumptions apply here as they did in the Air System. The mass balance over the control volume on the water side gives

$$\frac{dm^{ws}}{dt} = w^{ws}_{in} - w^{ws}_{out} - \dot{m}_l \tag{3.15}$$

Equation 3.15 states that the accumulation of water in the control volume, is equal to the mass flow of water into the control volume, w^{ws}_{in}, minus the mass flow out of the control volume, w^{ws}_{out}, and further subtracted by the amount of water evaporated into the air system, $\dot{m}_l$. The energy balance over the water system gives

$$\frac{d(m^{ws} \cdot H^{ws}_{out})}{dt} = w^{ws}_{in} \cdot H^{ws}_{in} - w^{ws}_{out} \cdot H^{ws}_{out} - Q^{ws}_{out} \tag{3.16}$$

H^{ws}_{in} is the enthalpy of the water entering the control volume, and H^{ws}_{out} is enthalpy of the water flowing out of the control volume. Q^{ws}_{out} is the heat transfer

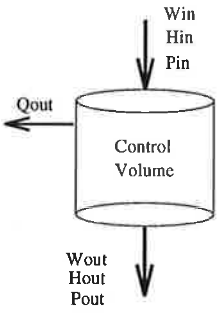

Figure 3.6: A Diagram over the Water System

to the contact area with the air side due to the mass of water being evaporated. The use of equation

$$\frac{d(m^{ws} \cdot H^{ws}_{out})}{dt} = \frac{dm^{ws}}{dt} \cdot H^{ws}_{out} + \frac{dH^{ws}_{out}}{dt} \cdot m^{ws}$$

gives the ODE

$$m^{ws} \cdot \frac{H^{ws}_{out}}{dt} = w^{ws}_{in} \cdot (H^{ws}_{in} - H^{ws}_{out}) + \dot{m}_l \cdot H_{out} - Q^{ws}_{out} \qquad (3.17)$$

Q^{ws}_{out} describes the heat flow rate out of the system. Since it is assumed that the *film* temperature is the same as that of the Water System the following shall hold true

$$Q^{ws}_{out} = \dot{m}_l \cdot H_{out} \qquad (3.18)$$

This is due to the assumption that a temperature gradient does not exist between the the bulk of the water flow and the area of contact between the two systems. It is off-course an ideal assumption. Then equation 3.17 will simplify to

$$m^{ws} \cdot \frac{H^{ws}_{out}}{dt} = w^{ws}_{in} \cdot (H^{ws}_{in} - H^{ws}_{out}) \qquad (3.19)$$

This is rather promising, since the water temperature of cooling tower used in the power plants and chemical industry does reach an *adiabatic cooling temperature* . In other words the Water System reaches a constant temperature. This is implemented in the **Omola** code.

The Film System Model

The contact area where down flowing water in the tower interacts with the up flowing air is modeled by the *film theory* [18, 19]. The film theory states that the two meeting flows are turbulent, except in the region of contact between the two fluids where the flow characteristics are *streamline*. Figure 3.7 shows the **Film System**. Film theory is only an approximation used to model the interaction of the two fluids at the contact surface, yet it is widely used. Several

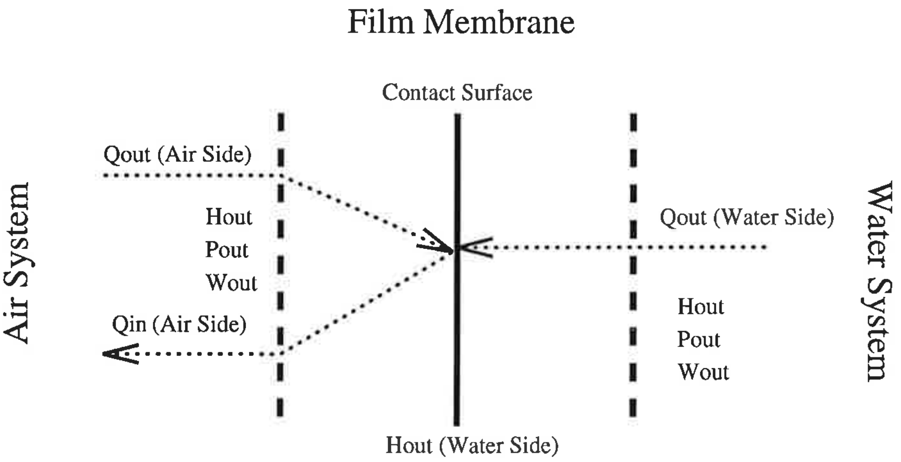

Figure 3.7: A Diagram over the Film System

other theories exist in this area of science. A mass balance over the Film System gives

$$\dot{m}_g = \dot{m}_l \tag{3.20}$$

This is an important assumption made that there is no accumulation of water at the surface area. All that leaves the Water System enters the Air System through the Film System. Furthermore, an important assumption to simplify the modeling is that the contact area between the air and the water has the same temperature as the Water System. This gives the following energy balance over the Film System.

$$Q^{as}_{out} + Q^{ws}_{out} = Q^{as}_{in} \tag{3.21}$$

This implies the energy balance over the Film System is in equilibrium and so that the energy flowing from the Air System due to the contact of the Air System with the Film System, Q^{as}_{out}, and the energy flowing from the Water System due to the contact of the Water System with the Film System, Q^{ws}_{out}, is equal to the energy flowing out of the Film System and into the Air System, Q^{as}_{in}.

There only remains to identify the terms in the equation 3.21 and a model of the Air-Water System can be made. The first term Q^{as}_{out} is due to convection and it can be described by the *Fourier Law* there the temperature gradient between the Air System and the contact surface in the Film System is the driving force. This is given by

$$Q^{as}_{out} = \alpha_G \cdot A \cdot (t_g - t_l) \tag{3.22}$$

α_G is the heat transfer coefficient and A is the area of contact between the two fluids in the Film System. t_g is the temperature of the air and t_l is the temperature of the water. The relation between temperature and enthalpy is given by $H = C_p \cdot t$, there C_p is the heat capacity of the respective fluidum.

The second term Q^{ws}_{out} is given by the equation

$$Q^{ws}_{out} = \dot{m}_l \cdot H^{ws}_{out} \tag{3.23}$$

This implies that the heat transfer from the Water System into the Film System is equal to the mass flow of the water multiplied by its enthalpy. $\dot{m}_l$ has the units $\frac{kg}{s}$ and H^{ws}_{out} has the units $\frac{kJ}{kg}$, so that the term Q^{ws}_{out} gives the effect $(\frac{kJ}{s})$ carried by the evaporated mass leaving the Water System and entering the Film System.

The last term is given by the equation

$$Q^{as}_{in} = \dot{m}_g \cdot (H^{evap}(t_{ws}) - H^{as}_{out}) \tag{3.24}$$

Equation 3.24 implies that the water evaporated from the surface of the contact area between the two fluids and entering the Air System should receive the equivalent of the heat of evaporation at the water temperature $H^{evap}(t_{ws})$ minus the enthalpy it shall have when the water has entered the Air System, $Hout^{as}_{out}$. This is due to the temperature gradient between the water surface and the air surface.

Now equation 3.21 has become

$$\alpha_G \cdot A \cdot (\frac{H^{as}_{out}}{C_{pg}} - \frac{H^{ws}_{out}}{C_{pl}}) + \dot{m}_l \cdot H^{ws}_{out} = \dot{m}_g \cdot (H^{evap}(t_{ws}) - H^{as}_{out}) \tag{3.25}$$

Solving the above equation for $\dot{m}_g$ gives

$$\dot{m}_g = \frac{\alpha_G \cdot A \cdot (\frac{H^{as}_{out}}{C_{pg}} - \frac{H^{ws}_{out}}{C_{pl}})}{H^{evap}(t_{ws}) - (H^{as}_{out} + H^{ws}_{out})} \tag{3.26}$$

This gives us the amount of water evaporated into the air system as the result of the simultaneous mass and heat transfer in the Film System. Equation 3.26 is an important conclusion in the proposed model of the **Humidification Tower**. H^{as}_{out} and H^{ws}_{out} are the states in the air side and the water side of the discretecized tower. These states change as a function of time due to the dynamics of the Air System and Water System; therefore, equation 3.26 is the relation required to model the simultaneous mass and heat transfer as a function of the dynamics of the tower, which will be discussed in the next section.

3.2.2 The Tray Model

Since a model has been proposed for the simultaneous mass and heat transfer in the counter current flow through the humidification tower, a model can be proposed for a **Tray Model**, which represents the interaction of the three subsystems: Air System, Water System and Film System. Figure 3.8 shows how the three subsystems are put together to build the Tray Model. The **Omola** code is

```
AirWat ISA Base::Model WITH

submodels:
% Submodels composing the Air and Water System
Air  ISA AirSys;
```

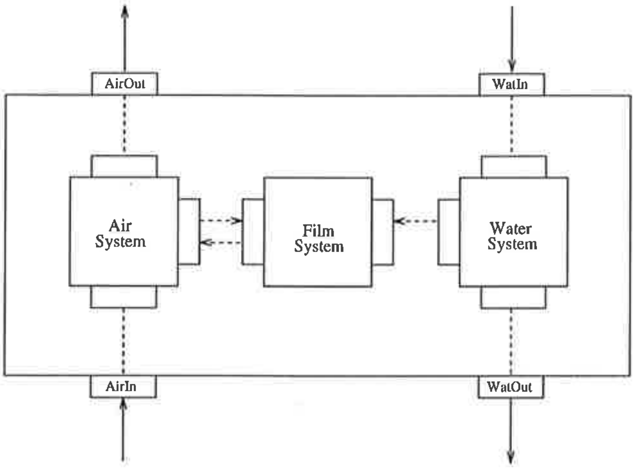

Figure 3.8: A Block Diagram over the Tray Model

```
Film ISA FilmSys;
Wat  ISA WatSys;

terminals:
% terminals of the Tray model to the outside
AirIn  ISA InCon;
WatIn  ISA InCon;
AirOut ISA OutCon;
WatOut ISA OutCon;

connections:
% The connections from the Air System
Air.Houtg AT Film.Houtg;
Air.Qout  AT Film.Qoutg;
Air.Qin   AT Film.Qing;
Air.mg    AT Film.mg;

% The connections from the Water System
Wat.Houtl AT Film.Houtl;
Wat.Qout  AT Film.Qoutl;
Wat.ml    AT Film.ml;

Equation:
Air.AirSysIn.Pin:=AirIn.Pin;
Air.AirSysIn.Hin:=AirIn.Hin;
Air.AirSysIn.Win:=AirIn.Win;
Wat.WatSysIn.Pin:=WatIn.Pin;
Wat.WatSysIn.Hin:=WatIn.Hin;
Wat.WatSysIn.Win:=WatIn.Win;
AirOut.Pout      := Air.AirSysOut.Pout;
AirOut.Wout      := Air.AirSysOut.Wout;
```

```
AirOut.Hout      := Air.AirSysOut.Hout;
WatOut.Pout      := Wat.WatSysOut.Pout;
WatOut.Wout      := Wat.WatSysOut.Wout;
WatOut.Hout      := Wat.WatSysOut.Hout;

END;
```

The three submodels **Air**, **Film**, and **Wat** described and modeled above are connected to model an ideal *cut* in the humidification tower.

3.2.3 Stabilization of the Tray Model

There are certain requirements made on the transient response of the Tray model based on the inputs which are the air and water enthalpies flowing into the system [2]. The Tray model has been introduced as a low order non-linear model where the constraints are on the mass and heat transfer across the film system. The ODE equations

$$\begin{aligned} m \cdot \frac{dH^{as}_{out}}{dt} &= w^{as}_{in} \cdot (H^{as}_{in} - H^{as}_{out}) - \dot{m}_g \cdot H^{as}_{out} + Q^{as}_{in} - Q^{as}_{out} \\ m^{ws} \cdot \frac{H^{ws}_{out}}{dt} &= w^{ws}_{in} \cdot (H^{ws}_{in} - H^{ws}_{out}) + \dot{m}_l \cdot H_{out} - Q^{ws}_{out} \end{aligned} \tag{3.27}$$

describe the dynamics of the enthalpies on the air and water side in the Tray model. By using equations 3.20- 3.26 the state equations become

$$\begin{aligned} m^{as} \cdot \frac{dH^{as}_{out}}{dt} &= -w^{as}_{in} \cdot H^{as}_{out} + w^{as}_{in} \cdot H^{as}_{in} - \frac{2\alpha_G A}{H^{evap} C_{pg}} {H^{as}_{out}}^2 + \frac{2\alpha_G A}{H^{evap} C_{pg}} H^{as}_{out} \cdot H^{ws}_{out} \\ m^{ws} \frac{H^{ws}_{out}}{dt} &= -w^{ws}_{in} \cdot H^{ws}_{out} + w^{ws}_{in} \cdot H^{ws}_{in} \end{aligned} \tag{3.28}$$

or rather

$$\begin{aligned} m^{as} \cdot \frac{dx_1}{dt} &= -w^{as}_{in} \cdot x_1 + w^{as}_{in} \cdot u_1 - \frac{2\alpha_G A}{H^{evap} C_{pg}} {x_1}^2 + \frac{2\alpha_G A}{H^{evap} C_{pl}} x_1 \cdot x_2 \\ m^{ws} \cdot \frac{dx_2}{dt} &= -w^{ws}_{in} \cdot x_2 + w^{ws}_{in} \cdot u_2 \end{aligned} \tag{3.29}$$

The state feedback control law given in equation 3.30 transforms the stabilization problem for the nonlinear system 3.29 into a stabilization problem for a controllable closed-loop system [4]. This state feedback control law cancels the nonlinearities of the system.

$$u_1 = \frac{a}{w^{as}_{in}} \left\{ \frac{{x_1}^2}{C_{pg}} - \frac{x_1 \cdot x_2}{C_{pl}} \right\} + \frac{v}{w^{as}_{in}} \tag{3.30}$$

There $a = \frac{2\alpha_G A}{H^{evap}}$. If v is chosen as $v = k_1 x_1 + k_2 x_2$ the following state equations are obtained.

$$\begin{aligned} m^{as} \cdot \frac{dx_1}{dt} &= -w^{as}_{in} \cdot x_1 + k_1 x_1 + k_2 x_2 \\ m^{ws} \cdot \frac{x_2}{dt} &= -w^{ws}_{in} \cdot x_2 + w^{ws}_{in} \cdot u_2 \end{aligned} \tag{3.31}$$

or in the matrix form the states of the Tray model can be represented as

$$\begin{pmatrix} m^{as}\dot{x}_1 \\ m^{ws}\dot{x}_2 \end{pmatrix} = \begin{pmatrix} -w_{in}^{as} + k_1 & k_2 \\ 0 & w_{in}^{ws} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ w_{in}^{ws} \end{pmatrix} u_2$$

This can be used to locate the eigenvalues of the system or place the desirable poles onto the system and thereby achieve the required transient response of the Tray model which is the building block of the humidification tower,**ABS**.

By introducing the controllability matrix

$$W_c = [B \quad AB]$$

the following result is obtained:

$$W_c = \begin{pmatrix} 0 & \frac{w_{in}^{as}}{m^{as}} \cdot \frac{w_{in}^{ws}}{m^{ws}} k_2 \\ \frac{w^{ws}}{m^{ws}} & \frac{w_{in}^{ws}}{m^{ws}} \cdot \frac{w_{in}^{ws}}{m^{ws}} \end{pmatrix} \tag{3.32}$$

Equation 3.32 implies that $detW_c \neq 0$ for $k_2 > 0$. Thereby for values of $k_2 > 0$ the designer may assign any closed loop poles that is as long as W_c has full rank. If the desired closed-loop characteristic equation is of the form

$$s^2 + p_1 s + p_2 = 0 \tag{3.33}$$

then the poles of the Tray model can be arbitrarily placed by putting

$$det(sI - A) = 0$$

and finding the characteristic equation. The characteristic equation of the system is

$$s^2 + (\frac{w_{in}^{as}}{m^{as}} + \frac{w_{in}^{ws}}{m^{ws}} - \frac{k_1}{m^{as}})s + \frac{w_{in}^{ws}}{m^{ws}} \cdot (-\frac{w_{in}^{as}}{m^{as}} + \frac{k_1}{m^{as}}) = 0 \tag{3.34}$$

By equating the equations 3.33 and 3.34 the following result can be obtained

$$\frac{w_{in}^{as}}{m^{as}} + \frac{w_{in}^{ws}}{m^{ws}} - \frac{k_1}{m^{as}} = p_1 \tag{3.35}$$

$$\frac{w_{in}^{ws}}{m^{ws}} \cdot (-\frac{w_{in}^{as}}{m^{as}} + \frac{k_1}{m^{as}}) = p_2 \tag{3.36}$$

3.2.4 The Tower Model

The **AirWat** represents the interaction of the three submodels on one so called tray. Several of these trays placed on top of one another should represent several *cuts* into the humidification tower at different heights. The **Omola** code which reflects this structure:

```
TowerModel ISA Base::Model WITH
% Five tray modell of the real thing.

submodels:
```

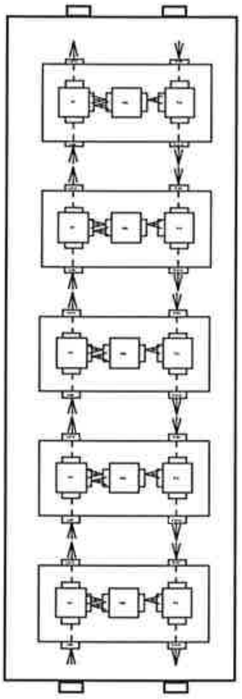

Figure 3.9: A Block Diagram over the Tower Model

```
% Submodels of the Tray Model composed
Tray1 ISA AirWat;
Tray2 ISA AirWat;
Tray3 ISA AirWat;
Tray4 ISA AirWat;
Tray5 ISA AirWat;

terminals:
% THe terminmal
TowerAirIn  ISA InCon;
TowerAirOut ISA OutCon;
TowerWatIn  ISA InCon;
TowerWatOut ISA OutCon;

Variables:
dTin,dTout ISA Variable;% Pinch studies over the AbsTower

connections:
Tray1.AirIn AT TowerAirIn;
TowerAirOut AT Tray5.AirOut;
Tray2.AirIn AT Tray1.AirOut;
Tray3.AirIn AT Tray2.AirOut;
Tray4.AirIn AT Tray3.AirOut;
Tray5.AirIn At Tray4.AirOut;
```

```
Tray1.WatIn AT Tray2.WatOut;
Tray2.WatIn AT Tray3.WatOut;
Tray3.WatIn AT Tray4.WatOut;
Tray4.WatIn AT Tray5.WatOut;
Tray5.WatIn AT TowerWatIn;
TowerWatOut AT Tray1.WatOut;

Equations:
dTin:=(Tray5.AirOut.Hout-Tray5.WatIn.Hin);
dTout:=(Tray1.AirIn.Hin-Tray1.WatOut.Hout);

END;
```

The five subunits of **AirWat** used in the model are numbered from one to five. A humidification tower is thereby constructed where the flow is a counter-current flow and the phenomena of simultaneous mass and heat transfer is also encapsulated. The air is observed to be humidified through the tower as shall be shown by the simulation results.

3.3 The Heat Exchanger Model

The dynamics of heat flow in processes is of great importance. The control of flow variables in order to provide necessary heat transport is a critical area in the design of process plants. A dynamic model describing the heat transport between different parts of a system provides the designer with the means for a faster and relatively accurate decision-making step. Parameters effecting the thermodynamic efficiency of the plant is accessed by simulation.

3.3.1 The Heat Exchanger Model

The heat exchanger model shows the dynamic transport of heat through a contact surface of a counter-current flow of two fluids. This same model is going to represent both the **Rec** and the **IC** units in the Humid Air Gas Turbine model library.

The recuperator is placed at the outlet of the expander so that the hot exhaust gas leaving the expander can exchange heat with the cool humidified air entering the cold side of the heat exchanger. This design will allow the plant to operate by less fuel.

The intercooler is used to exchange heat between the hot air at the outlet of the low compressor with the water circulation system. This design effects also the performance of the power plant. Intercooling has both thermodynamic and mechanical advantages [21].

Figure 3.10 shows a block diagram of a heat exchanger. This diagram also shows the interaction of the heat exchanger with the outside world. The inflow of warm fluid on the *Warm Side* is termed **HeatExWarmSideIn**. This is also used as a terminal name to connect the heat exchanger to the other units. In the same way the outflowing *warm side* fluid is termed **HeatExWarmSideOut**. On the *cold side* the same terminology is used as shown in the diagram.

The modeling or the mathematical realization of the heat exchanger requires that a decomposition is made on the contact surface between the two

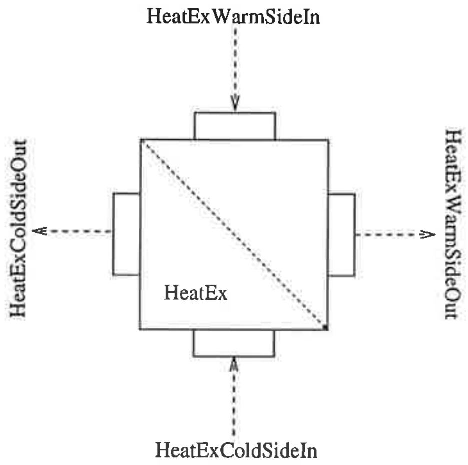

Figure 3.10: The Heat Exchanger Block Diagram

fluids. Figure 3.11 shows the decomposed heat exchanger. **Q** represents the heat transferred over to the ***cold side*** and the arrow in the figure shows the direction of the heat transfer.

A mass and energy balance over the cold and warm side of the heat exchanger with respect to the dynamics gives a set of ODE equations. The mass balance over the *warm side*, **ws**, is

$$\frac{dm^{ws}}{dt} = w_{in}^{ws} - w_{out}^{ws} \tag{3.37}$$

and on the *cold side*, **cs**, mass balance becomes

$$\frac{dm^{cs}}{dt} = w_{in}^{cs} - w_{out}^{cs} \tag{3.38}$$

Both $\frac{dm^{ws}}{dt}$ and $\frac{dm^{cs}}{dt}$ are accumulation terms on each side of the heat exchanger. The energy balance over the cold side and warm side also gives:

$$\frac{d(mH_{out}^{ws})}{dt} = w_{in}^{ws} H_{in}^{ws} - w_{out}^{ws} H_{out}^{ws} - Q \tag{3.39}$$

$$\frac{d(mH_{out}^{cs})}{dt} = w_{in}^{cs} H_{in}^{cs} - w_{out}^{cs} H_{out}^{cs} + Q \tag{3.40}$$

Q is the heat transferred between the counter flowing fluids on the contact surface. It is given by the relation in equation 3.41

$$Q = U \cdot A \cdot \frac{\Theta_1 - \Theta_2}{\ln \frac{\Theta_1}{\Theta_2}} \tag{3.41}$$

U is the heat transfer coefficient, A is the contact area between the two fluids. $\Theta_1 = H_{in}^{ws} - H_{out}^{cs}$ and $\Theta_2 = H_{out}^{ws} - H_{in}^{cs}$. The expression below is sometimes used to show the relation.

$$Q = U \cdot A \cdot \Delta T_{ln} \tag{3.42}$$

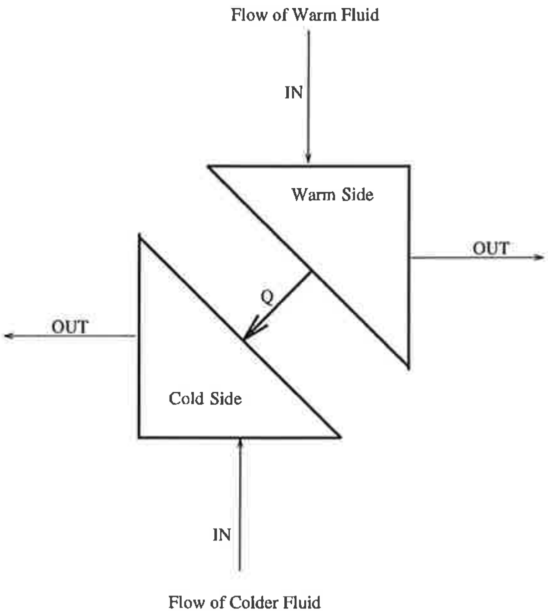

Figure 3.11: The Decomposition of the Recuperator

there ΔT_{ln} is called the ***logarithmic temperature difference.***
By defining θ in this way the program further provides the user with the means of a ***pinch study*** as a function of time. The transient characteristics of the heat transfer can be studied, but it should be pointed out that only a more complex model can provide the engineer with sound data about the transients of the heat transfer. ***Pinch studies*** is a very common tool in the optimization of power plants.

The Omola code for the Recuperator is:

```
HeatEx ISA Model WITH
Parameters:
Kc ISA Parameter;
Kw ISA Parameter;
U  ISA Parameter;
A  ISA Parameter;
m  ISA Parameter WITH default:=500;END;
Cpg ISA Parameter WITH default:=1.4;END;

Terminals:
HeatExWarmSideIn  ISA InCon;
HeatExWarmSideOut ISA OutCon;
HeatExColdSideIn  ISA InCon;
HeatExColdSideOut ISA OutCon;

Variables:
Diff1   ISA Variable;
Diff2   ISA Variable;
```

```
Theta1  ISA Variable WITH initial:=5000;END;
Theta2  ISA Variable WITH initial:=4000;END;
Q       ISA Variable ;

E1      ISA Variable;
E2      ISA Variable;
E3      ISA Variable;
E4      ISA Variable;

Equations:
HeatExWarmSideOut.Pout:=HeatExWarmSideIn.Pin-Kw*SQR(HeatExWarmSideOut.Wout);
HeatExWarmSideOut.Wout:=HeatExWarmSideIn.Win;
HeatExColdSideOut.Pout:=HeatExColdSideIn.Pin-Kc*SQR(HeatExColdSideOut.Wout);
HeatExColdSideOut.Wout:=HeatExColdSideIn.Win;
E1=HeatExColdSideIn.Win*HeatExColdSideIn.Hin;
E2=HeatExColdSideOut.Wout*HeatExColdSideOut.Hout;
E3=HeatExWarmSideIn.Win*HeatExWarmSideIn.Hin;
E4=HeatExWarmSideOut.Wout*HeatExWarmSideOut.Hout;

Diff1 =(E1-E2);
Diff2 =(E3-E4);

m*HeatExColdSideOut.Hout'=Diff1+Q;
m*HeatExWarmSideOut.Hout'=Diff2-Q;

Theta1=HeatExWarmSideIn.Hin-HeatExColdSideOut.Hout;
Theta2=HeatExWarmSideOut.Hout-HeatExColdSideIn.Hin;

Q = U*A/Cpg*(Theta1-Theta2)/LN(abs(Theta1/Theta2));

END;

RecModel     ISA HeatEx WITH
% The recuporator model with modified parameters
  U  ISA Parameter WITH default:=0.05;END;%Heat transfer coefficient
  A  ISA Parameter WITH default:=2000;END;%The contact area
  Kc ISA Parameter WITH default:=10;END;% The friction coeff. cold side
  Kw ISA Parameter WITH default:=10;END;% The friction coeff. warm side
END;
```

The recuporator is a heat exchanger and it is modeled here as **HeatEx**. Then RecModel inherits the model HeatEx with its own set of parameters. The equations discussed earlier are implemented in the Omola syntax and the model is again connected to other units by the terminals **UnitConIn** and **UnitConOut**. The Rec unit is a major source of pressure loss so it shall be included in the model. A model often used to describe the pressure loss as a function of the flow is:

$$\Delta P = \zeta \frac{\rho v^2}{2} \tag{3.43}$$

This can be more easily implemented by using $\Delta P = \kappa w_{out}^2$, there **w** is the mass flow in (kg/s) on each side of the heat exchanger. A look at the Omola code shows the parameters **Kc** and **Kw**. These can be used to manipulate the pressure loss in the unit as a function of flow.

3.4 The Tank System Model and Control

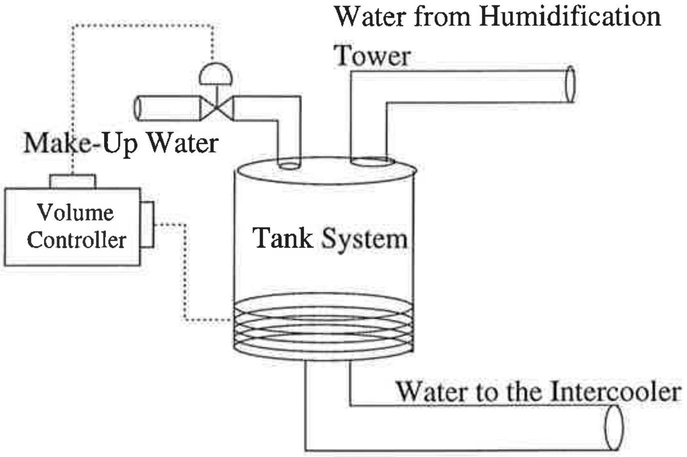

Figure 3.12: The Tank System and Volume Controller

The mass balance over the tank shown in figure 3.12 is

$$\frac{dm}{dt} = w_{in} + w_r - w_{out} \tag{3.44}$$

An ideal mix is assumed and that the enthalpy of the water can be taken as equal to the internal energy of the water. This implies that the mass of water accumulated per time unit is equal to the mass inflow of the water leaving the humidification tower, w_{in}, plus the amount of *make-up water* w_r subtracted by the amount of water leaving the tank towards the intercooler w_{out}. The energy balance over the tank system gives

$$\frac{d(m \cdot H_{out})}{dt} = w_{in} \cdot H_{in} + w_r \cdot H_r - w \cdot H_{out} \tag{3.45}$$

where H_r is the enthalpy of the *make-up water* added to the tank . The ordinary differential equation derived from 3.45 is

$$m \cdot \frac{dH_{out}}{dt} = w_{in} \cdot (H_{in} - H_{out}) + w_r \cdot (H_r - H_{out}) \tag{3.46}$$

The **Omola** code is

```
Tank ISA Model WITH

Terminals:
```

```
% The terminals or the abstract surface of the Tank System
InFlow  ISA InCon;
OutFlow ISA OutCon;

Parameter:
   Vref     ISA Parameter WITH default:=50  ;END;% The volume ref. value
   Ro       ISA Parameter WITH default:=1000;END;% The density of water
   Hr       ISA Parameter WITH default:=100 ;END;% Water Enthalpy
   h        ISA Parameter WITH default:=0.5 ;END;% The samplings period
   ti       ISA Parameter WITH default:=80;END;  %Integrator
   PressEq  ISA Parameter WITH default:=12E5;END;% Pressure Equaliser
   MassFlow ISA Parameter WITH default:=100 ;END;% To the intercooler
   k        ISA Parameter WITH default:=10  ;END;% The prop. controller
   A        ISA Parameter WITH default:=50  ;END;%tank area

Variables:
m ISA Variable WITH initial:=1000;END;
V ISA Variable WITH initial:=50  ;END;
P ISA Variable; % Just to test the pressure loss

Equations:
   OutFlow.Pout  := PressEq;
   P             := InFlow.Pin;
   OutFlow.Wout  := MassFlow;
   V             := m/Ro;
   m'             = InFlow.Win + Wr - OutFlow.Wout;
   m*OutFlow.Hout'= InFlow.Win*(InFlow.Hin-OutFlow.Hout)+Wr*(Hr-OutFlow.Hout);
Events:
 inte, e, u ,Wr TYPE DISCRETE REAL;
  Init, Sample ISAN Event;
  ONEVENT Init OR Sample DO
    new(e) := Vref - V;
    new(inte) := inte + k*e*h/ti;
    new(u) := k*new(e) + inte;
    SCHEDULE(Sample,h);
  END;
  Wr:= u;
END;
```

On the event of initializing the simulation program,**Init** or the sampling period **Sample**, the value of the error,e, is determined and a new control signal w_r is calculated [3, 8]. This keeps the volume of the tank constant which is necessary to make up for amount of water evaporated in the humidification tower.

Chapter 4

Thermodynamics of Power Plants

The study and optimization of thermal efficiency is a classical area of interest for power engineers [11, 12, 14]. Although the optimization of the humid air gas turbine cycle is out of the scope of this thesis, it is however important to provide the means for the model user to compare and validify the built model with thermodynamic data from running plants.

The use of relative thermodynamics from different power plants eases the need to use data based on real-time 'spot-values'. This is acceptable since power plants with different complexities, feul intake, useful heat to work ratio, size, power demand, etc, etc, can have the same thermal efficiency but enormously different 'spot-values'. So, a study of the thermodynamics of the **HAT cycle** provides the user with the means of establishing the thermal efficiencies of the power plant, and then manipulating the model parameters to achieve the desired result.

4.1 Comparative Thermodynamic Performance

4.1.1 Efficiency and Energy Utilization Factor

The heat recuporated in the system is defined as useful heat. λ_{CG} is then the quote between the useful heat recuporated and the work produced by the system.**CG** stands for co-generation since both work and useful heat is being produced in the power plant.

$$\lambda_{CG} = \frac{Q_u}{W} \tag{4.1}$$

An overall energy efficiency factor is defined as

$$\eta_0 = \frac{W}{F} \tag{4.2}$$

which is the work to fuel ratio. Equation 4.3 is called the Energy Utilization Factor. This is the quote between the net work produced and the useful heat recuporated to the fuel used to run the open cycle plant discussed.

$$EUF = \frac{W + Q_u}{F} \tag{4.3}$$

4.1.2 Artificial Thermal Efficiency

The energy supply to the plant is reduced by the useful heat recuporated i.e Q_u. If a separate plant were to produce the heat load Q_u separatedly in a *heat only* boiler of efficiency η_B^H, there the feul cost will be $\frac{Q_u}{\eta_B^H}$. Then the feul saved will be included in a alternative criterion called **Artificial Thermal Efficiency** [10], η_a. This criterium is given by

$$\eta_a = \frac{W}{F - \frac{Q_u}{\eta_B^H}} \tag{4.4}$$

This as a function of our earlier criteria becomes

$$\eta_a = \frac{\eta_0^{CG}}{1 - \frac{Q_u}{\eta_H^B \cdot F}} \tag{4.5}$$

where η_0^{CG} is the overall efficiency of the CHP plant. η_B^H is normally taken as 0.90.

4.1.3 Feul Energy Savings Ratio

This criterium is defined as an assessment of a power plant with variable thermodynamic parameters such as inlet and reheat conditions ,back pressure,etc. The generated heat and work in a co-generation power plant can be compared with a power plant only producing power and a power plant only producing heat. A Combined Heat and Power, CHP, plant is therefore compared with a conventional electric power station of overall efficiency η_0^C and a *heat only* boiler of efficiency η_B^H. Then the feul energy saved is

$$\Delta F = \frac{Q_u}{\eta_H^B} + \frac{W}{\eta_0^C} - F \tag{4.6}$$

and the Feul Energy Saving Ratio **FESR** is defined as the ratio of the saving to the feul energy required in the two above mentioned conventional plants

$$FESR = \frac{\Delta F}{\frac{Q_u}{\eta_B^H} + \frac{W}{\eta_0^C}} \tag{4.7}$$

$$FESR = 1 - \frac{\frac{\eta_0^C}{\eta_0^{CG}}}{1 + \frac{\lambda_{CG} \cdot \eta_0^C}{\eta_B^H}} \tag{4.8}$$

λ_{CG} and η_0^{CG} is defined earlier and η_0^H is assumed to be 0.90. η_0^C is normally taken as 0.40.

4.1.4 Incremental Heat Rate

The incremental heat rate assumes that the feul energy is broken down into one supplying the heat load and the other the power generation. Then it is assumed that the amount of feul used to produce power is the **balance** of the feul needed to produce heat. In other words IHR_{CG} is given by the equation

$$F_{CG} = \frac{Q_u^{CG}}{\eta_B^{CG}} + IHR_{CG}W_{CG} \tag{4.9}$$

where IHR_{CG} is a coefficient in the equation. This criterium is introduced by Porter and Mastanaiah in analyzing and assessing the economics of a co-generation power plant. Solving for IHR_{CG} gives

$$IHR_{CG} = \frac{F_{CG}}{W_{CG}} - \frac{Q_u^{CG}}{\eta_B^{CG} \cdot W_{CG}} \tag{4.10}$$

which is

$$IHR_{CG} = \frac{1}{\eta_0^{CG}} - \frac{\lambda_{CG}}{\eta_B^{CG}} \tag{4.11}$$

This is IHR_{CG} as a function of our earlier criteria.

4.2 Comparative Plants

It is useful to have thermodynamic data from similarly high performance power plants inorder to compare data from the simulation results. Here, four different installations used in the industry shall be introduced [12, 13, 14]. These four are

- Extraction and Condensing Plant
- Back Pressure Plant
- Gas Turbine with Waste Heat Recuporation
- Combined Cycle(Back Pressure Steam Turbine/Gas Turbine)

This list of different power plants shall be introduced and their respective thermodynamic data given.

4.2.1 The Extraction and Condensing Plant

Figure 4.1 shows an Extraction and Condensing plant which is a double-stage steam turbine plant where extraction occurs to warm feed-water entering the boiler. The process steam is condensed by the condenser and a circulation pump is used to increase the pressure on the water side leaving the condenser. This plant has the capability to be controlled to obtain variable process steam requirements or power demand. The process steam requirements are met by extracting steam at a certain point from the steam turbine. Table 4.2.1 lists the thermodynamic data from a running extraction and condensation plant.

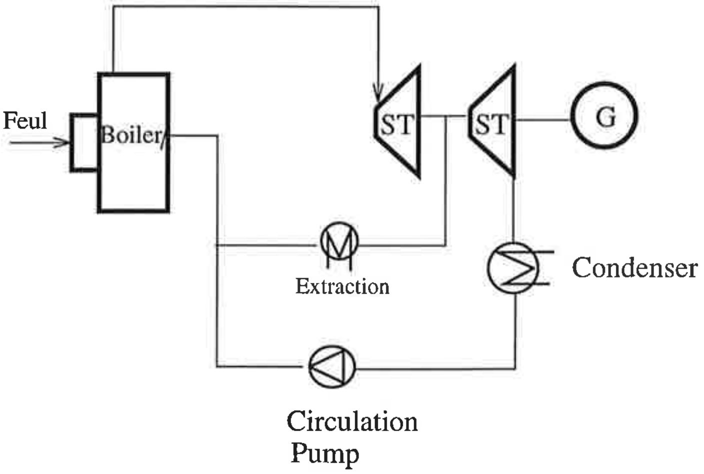

Figure 4.1: The Extraction and Condensing Plant

Drift Data: The Extraction and Condensing Plant

Useful Heat to Work Ratio:	λ_{CG}	0.26
Overall Efficiency:	η_O	0.38
Energy Utilization Factor:	EUF	0.48
Artificial Thermal Efficiency:	η_a	0.43
Feul Energy Saving Ratio:	FESR	0.057
Incremental Heat Rate:	IHR_{CG}	2.33

4.2.2 Back Pressure Plant

The back pressure plant,fig. 4.2, is a single-stage steam turbine unit where the steam rates are controlled, inorder to achieve the required power demand from the generator. The pressure over the steam turbine is *set*. A comparisson with the extraction and condensation plant shows that it has a lower overall efficiency but a greater useful heat to work ratio.

Drift Data: The Back Pressure Plant

Useful Heat to Work Ratio:	λ_{CG}	2.40
Overall Efficiency:	η_O	0.25
Energy Utilization Factor:	EUF	0.85
Artificial Thermal Efficiency:	η_a	0.75
Feul Energy Saving Ratio:	FESR	0.235
Incremental Heat Rate:	IHR_{CG}	1.33

4.2.3 Gas Turbine with Waste Heat Recuporation

This plant is a gas turbine power plant fitted with a waste heat recuporator. The proposed model for the HAT cycle will produce data which resumbles this type of plant.

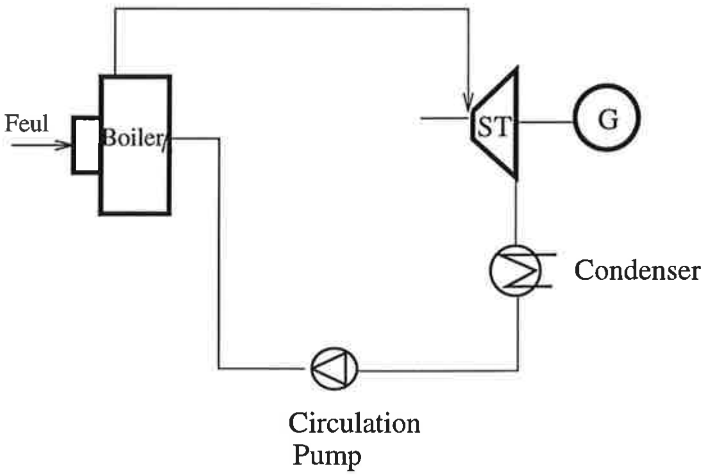

Figure 4.2: The Back Pressure Plant

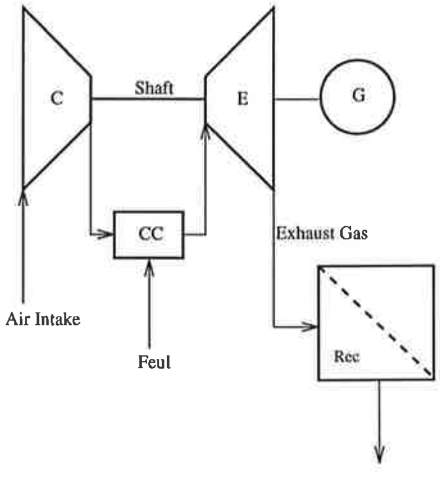

Figure 4.3: The Gas Turbine and Waste Heat Recuporator

Drift Data: The Gas Turbine and Waste Heat Recuporator

Useful Heat to Work Ratio:	λ_{CG}	1.83
Overall Efficiency:	η_0	0.3
Energy Utilization Factor:	EUF	0.85
Artificial Thermal Efficiency:	η_a	0.77
Feul Energy Saving Ratio:	FESR	0.265
Incremental Heat Rate:	IHR_{CG}	1.3

4.2.4 Combined Cycle

This is the most highly rated power plant used in the field of cogeneration. It is a combination of a gas turbine unit and a back pressure power plant. The technology for the construction of this power plant is now conventional and today it is a common challange in the field of power engineering to compare this system to the Humid Air Gas Turbine Cycle.

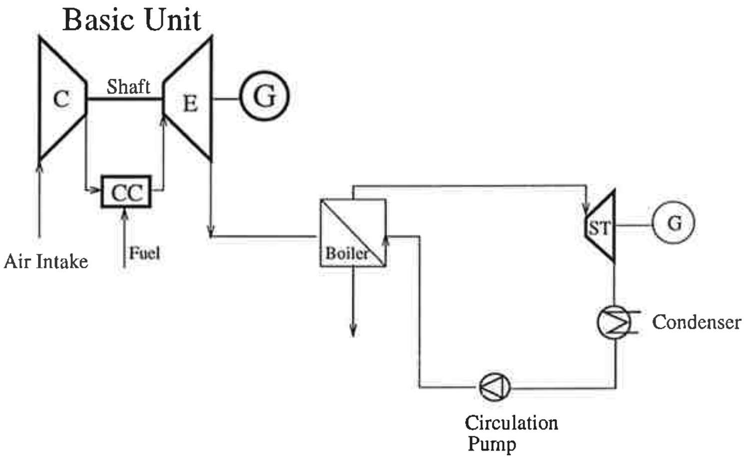

Figure 4.4: The Combined Cycle

Drift Data: The Combined Cycle

Useful Heat to Work Ratio:	λ_{CG}	1.05
Overall Efficiency:	η_0	0.40
Energy Utilization Factor:	EUF	0.82
Artificial Thermal Efficiency:	η_a	0.75
Feul Energy Saving Ratio:	FESR	0.318
Incremental Heat Rate:	IHR_{CG}	1.33

Chapter 5

Validation by Simulation Results

5.1 The Dynamics of The Humidification Tower

Figure 5.1 shows the change in the enthalpy of the Air System as a function of the Water System enthalpy. This simulation is carried out for only one Tray Model by **Simnon** [1]. The square signal is representative of the variation in the enthalpy of the water entering the humidification tower, while the water flow is kept constant. The air enters the tray after the low compressor,**LC**. This is the result of the state feedback control law **3.30** and discussions in section 3.2.3. The control law is showing satisfactory result as the enthalpy of the Air System is being controlled by the desirable control variable water enthalpy. The **Simnon** code for this simulation is listed in the Appendix.

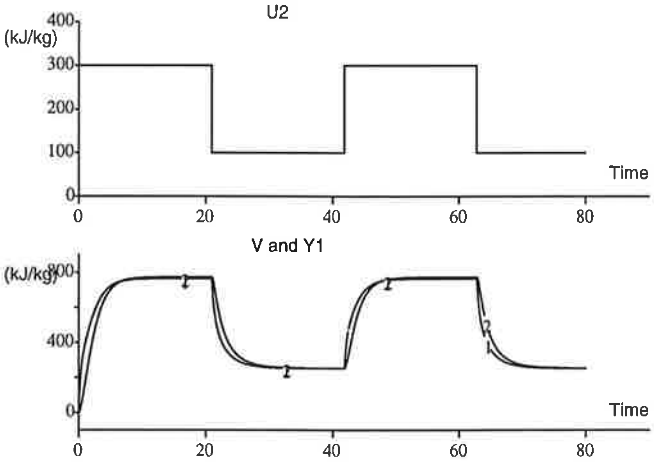

Figure 5.1: The Air System Enthalpy with Water Enthalpy as Input

Figure 5.2 is showing the enthalpy profile of the different trays in the humid-

---

[1] Simnon is a trademark of Department of Automatic Control, Lund, Sweden

ification tower,**ABS**. The tray indexed 1 is at the bottom of the tower where the air flows in. This an open-loop simulation result as compared to that of the closed-loop system shown for only one tray in figure 5.1.

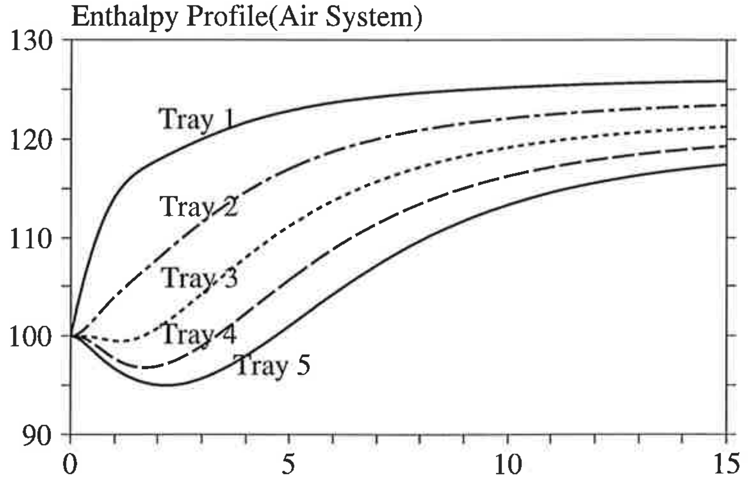

Figure 5.2: The Enthalpy Profile of the Air System in the Tower

Figure 5.3 shows the change in enthalpy of the water system as the function of the input signal water enthalpy. The square input signal represents the variation in the water enthalpy leaving the intercooler and entering a single Tray Model. The trays enthalpy on the water side is satisfactorily controlled.

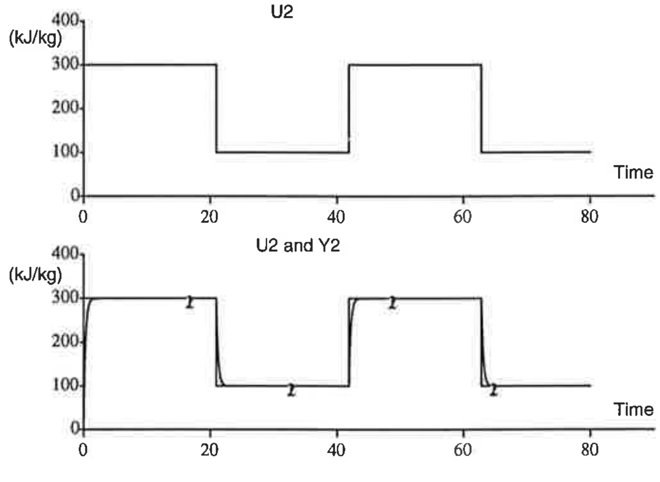

Figure 5.3: The Enthalpy Profile of the Water System

The final enthalpy of the trays in the humidification tower converge into a single value as a function of time, figure 5.4. This is due to the large amount of energy flowing in and out of the control volume here modeled as the water system, so that in comparison, the loss of energy due to the mass of water being evaporated at the film membrane does not effect the final value of the water system enthalpy in a significant manner. This profile can be altered by increasing the process parameters that effect the heat and mass transfer across the film membrane . The water flow for this simulation is 100 $\frac{kg}{s}$ as compared to the average flow of evaporated water between 2 to 2.5 ($\frac{kg}{s}$) figure 5.5.

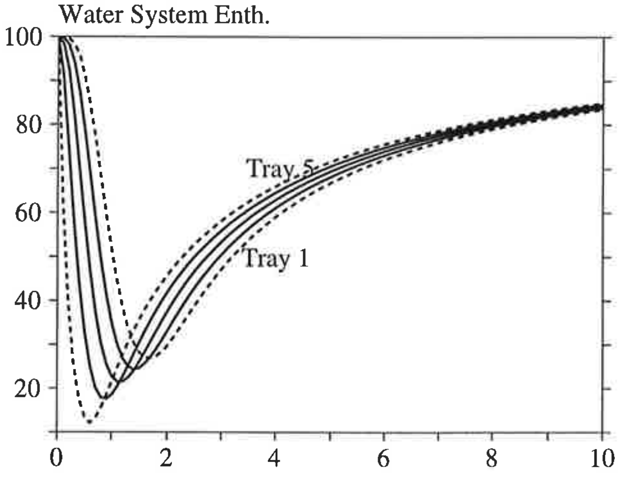

Figure 5.4: The Enthalpy Profile of the Water System in The Tower

The mass transfer on each tray is the result of the states of the surrounding air and water system. Figure 5.5 shows how the mass flow through the film system varies on each tray and how they reach a stationary value as a function of time.

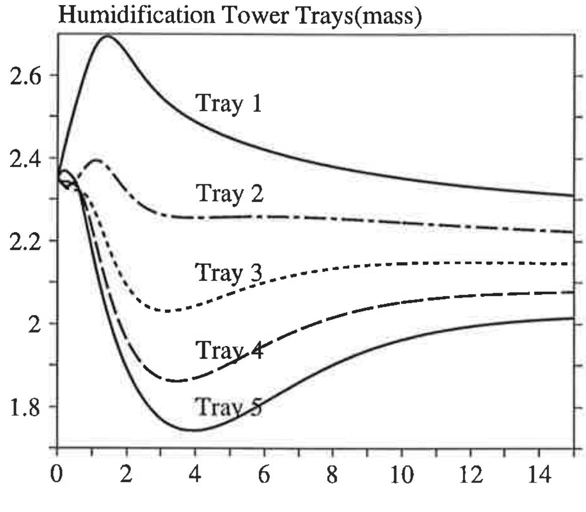

Figure 5.5: The Evaporated Mass of Water on Each Tray Model in The Tower

The mass transfer in this proposed model of the humidification tower is among other parameters a function of the ***heat transfer coefficient***, α_G, and the area of contact between the two fluids, A. These two parameters depend on the design specifics of the manufacturer, and may vary enormously depending on the use and size of the tower. Figure 5.6 shows how the total increase of mass flow in the tower, due to humidification, is a function of the choice of this parameter, $\alpha_G \cdot A$. There is an increase in the total mass of water being evaporated as the value of $\alpha_G \cdot A$ is increased.

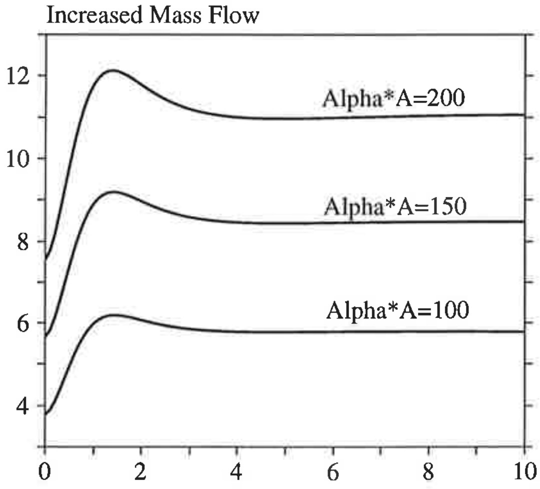

Figure 5.6: A Process Parameter: *Heat Transfer Coefficient* α_G

5.2 The Heat Exchanger

The recuperator, **Rec**, is a dynamic model of a heat exchanger. The dynamics of the recuperator are nonlinear so that a *phase plane analysis* over the model gives a better understanding of its behaviour.

5.2.1 Phase Plane Analysis over the Recuperator

Figure 5.7 shows the trajectories of the states in the recuperator model. The two states of the model are the enthalpy of the exhaust gas leaving the recuperator on the *warm side* and the enthalpy of the gas leaving the recuperator on the *cold side.* The *cold side* of the recuperator is connected to the combustion chamber, while the *warm side* flows out into the atmosphere.

The figure shows that if the initial values given to the state on the *warm side* is between $640 - 800\frac{kJ}{kg}$ and the initial values given to the state on the *cold side* is between $450 - 700\frac{kJ}{kg}$ the trajectories of the states in the model will converge onto a stable focus point at about $(660, 600)$. This co-ordinates of the stable focus point is strongly dependent on choice of the process parameters such as the fuel flow and air flow in the power plant.

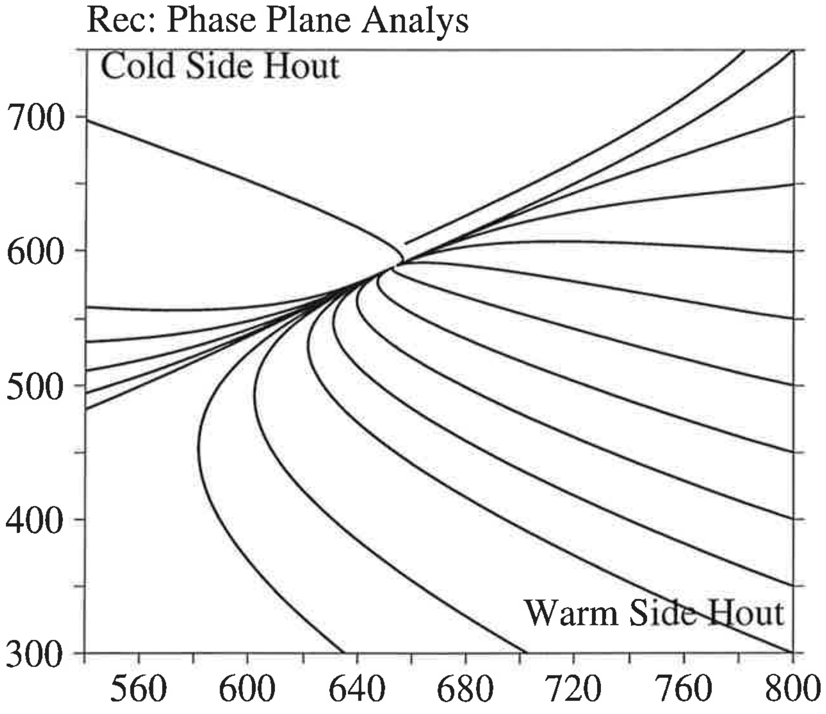

Figure 5.7: The Phase Plane Portrait

5.3 The Comparative Thermodynamics of the Cycle

Several different methods exist to judge the thermodynamic performance of a power plant. Some of these different methods were discussed in chapter 4. In this section, the following criteria shall be discussed to determine and access the thermodynamic performance of the modeled HAT cycle:

λ_{CG} Defined as the Useful Heat to Work Ratio

η_0 The Overall Efficiency defined as the Work to Fuel Ratio

FESR The Fuel Energy Saving Ratio

5.3.1 Comparative Thermodynamics as a Function of Pressure Ratio over the Compressor

The useful heat to work ratio, λ_{CG}, of the HAT cycle is studied. It is a function of the pressure ratio over the compressor, 5.8. If the ratio is 3 : 1 then there is more useful heat produced in the plant as compared to the work output at the generator. λ_{CG} decreases as the pressure ratio increases. This is due to the greater output at the generator as compared to the useful heat recuporated in the power plant. In the power plants introduced in chapter 4, the ***Back Pressure Plant*** had the highest ratio, $\lambda_{CG} = 2.4$. The ***Combined Cycle*** had a $\lambda_{CG} = 1.05$.

This performance can be obtained by a HAT cycle if a pressure ratio of 3 : 1 is chosen over the compressor system.

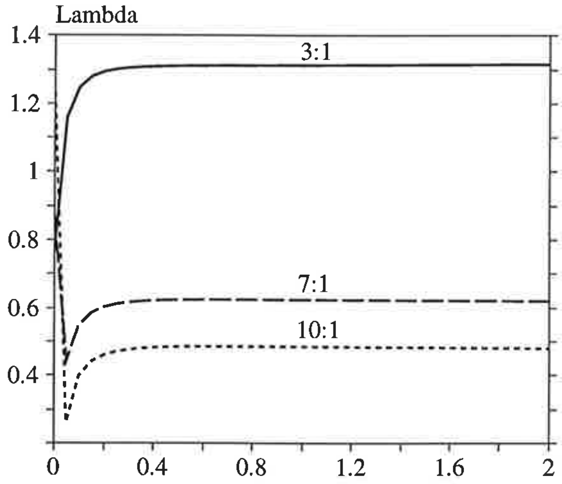

Figure 5.8: Useful Heat to Work Ratio, λ_{CG} : Different Pressure Ratios

The overall efficiency which is defined as the work to fuel ratio, η_0, increases as the pressure ratio over the compressor of the HAT cycle increases, figure 5.9. The HAT cycle with a pressure ratio of 3 : 1 has an lower over all efficiency than the *Combined Cycle*, but its η_0 is approximately equal to that of a *Gas Turbine and Waste Heat Recuperator Plant.* This criteria can easily be increased for the HAT cycle by increasing the pressure ratio over the compressor, which will result in the best work output compared with the fuel intake in the plant.

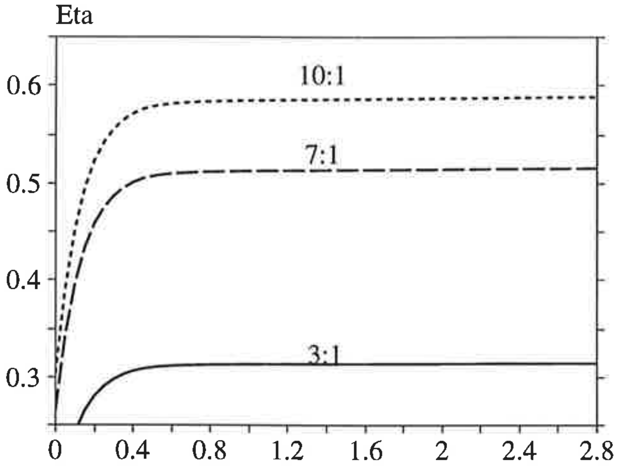

Figure 5.9: Work to Fuel Ratio, η_0 : Different Pressure Ratios

The Fuel Energy Saving Ratio, **FESR**, is the criteria for the fuel consumption of a co-generation power plant as compared to the consumption of fuel by two different plants, one producing work at the generator and the other a boiler producing useful heat. **FESR** is procentage fuel saved by using the cogeneration alternative. Figure 5.10 shows the **FESR** as a function of pressure ratio over the compressor system in the HAT cycle. The higher this pressure ratio becomes the higher the amount of fuel saved by the system. In power plants the *Combined Cycle* has the highest fuel energy saving ratio, **FESR** $= 0.318$, and the *Extraction and Condensation Plant* the lowest **FESR** $= 0.057$. The values obtained by the modeled HAT cycle surpasses that of the Combined Cycles for a $7:1$ ratio over the compressor.

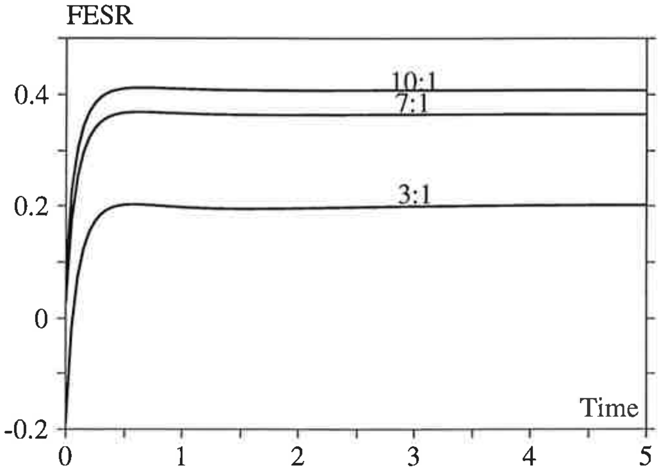

Figure 5.10: Fuel Energy Saving Ratio, $FESR$: Different Pressure Ratios

5.3.2 Comparative Thermodynamics as a Function of Recuperator Area

Figure 5.11 shows the useful heat to work ratio, λ_{CG}, in the HAT cycle for a 7:1 pressure ratio over the compressor system as a function of increasing recuperator areas. The figure shows an increase in the value of λ_{CG} as the area of the recuperator area increases. this means that a greater amount of useful heat is being recuperated into the system. The value obtained for a recuperator of area $1750m^2$ is $\lambda \approx 1$, which is equal to that of a *Combined Cycle* of similar magnitude and output.

The overall efficiency of the HAT cycle as a function of different recuperator areas is shown in figure 5.12. The work to fuel ratio, η_0, here defined as the overall efficiency, does not change significantly as a function of recuperator area, even if the value for $\eta_0 \approx 0.50$ is the largest among the introduced power plants. An overall efficiency of 50% is larger than that of the *Combined Cycle* which is only 40%.

The fuel energy saving ratio, **FESR**, as a function of different recuperator areas is shown in figure 5.13. The values of **FESR** increases successively as the recuperator area is increased, which implies that more fuel is being saved. This can also be interpreted as more energy is being produced by the same amount of fuel being injected into the plant, when a larger recuperator area is used. A

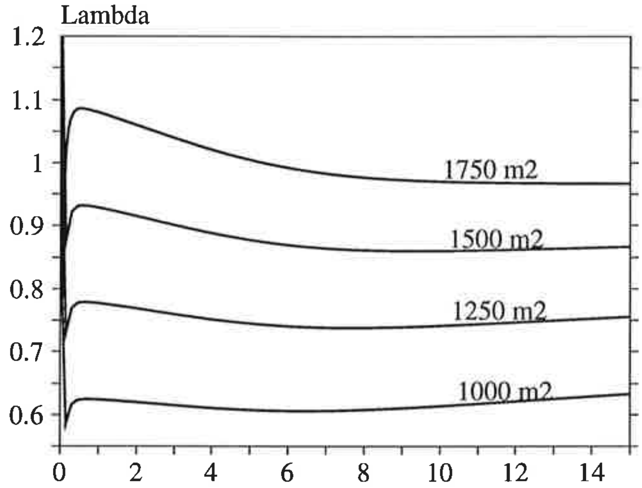

Figure 5.11: Useful Heat to Work Ratio, λ_{CG} : Different Recuporator Areas

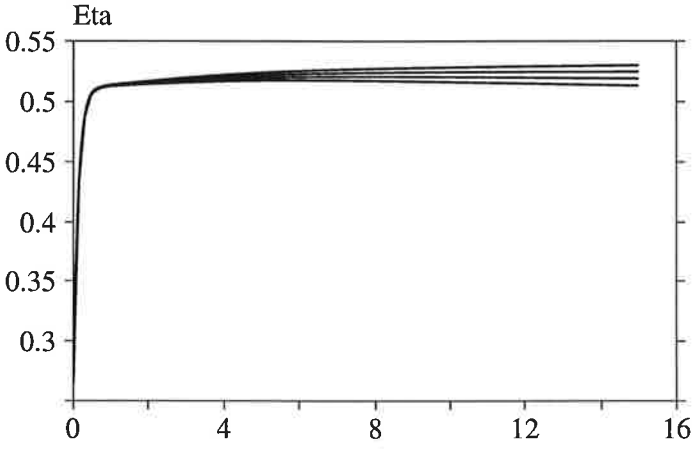

Figure 5.12: Work to Fuel Ratio, η_0 : Different Recuporator Areas

recuperator of area 1000 m^2 has approximately the same **FESR** as a *Combined Cycle*. So, an increase in the recuperator area is an energy saving measure.

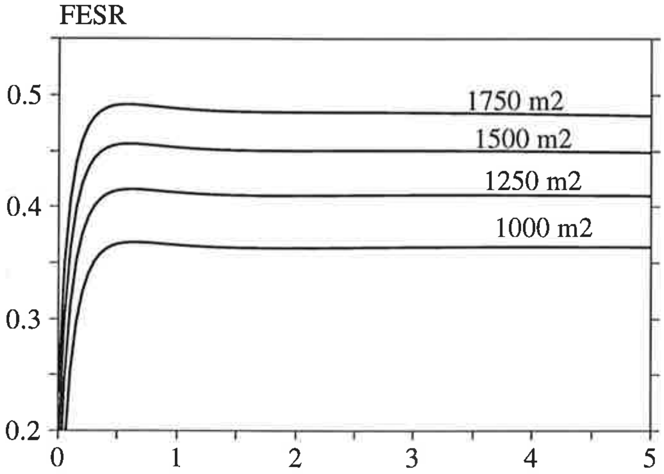

Figure 5.13: Fuel Energy Saving Ratio, $FESR$: Different Recuperator Areas

5.4 Generator Output

The power output of the HAT cycle is defined as the work produced by the expander subtracted by the work consumed by the compressors. Figure 5.14 shows the power output of the HAT cycle as a function of the different pressure ratios over the compressor system. The figure shows that this value increases as the pressure ratio over the compressor system in the plant is increased. A constant fuel-to-air ratio is used in all these simulations. Thereby, a fuel-to-air ratio of 1:40 gives a power output of approximately 27 MW in the HAT cycle when a pressure ratio is 7:1, and the recuperator area is $1000m^2$.

5.5 The Tank System

The water circulation system in the HAT cycle includes a water tank, figure 3.12 on page 30. The water streaming out of the humidification tower is pored into the water tank. In this proposed model of the HAT cycle a volume controller is implemented on the tank. Figure 5.15 shows the water in the tank being controlled to carry $50m^3$ of water and the control signal is the *make-up water* flow. This method also compensates for the water being evaporated in the humidification tower.

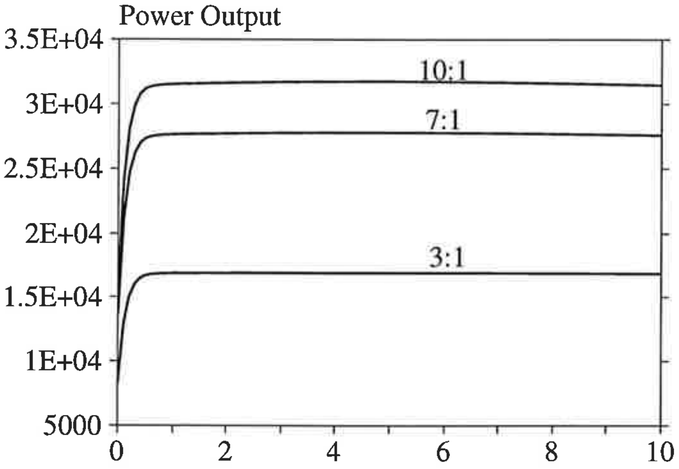

Figure 5.14: The Power Output: Different Pressure Ratios

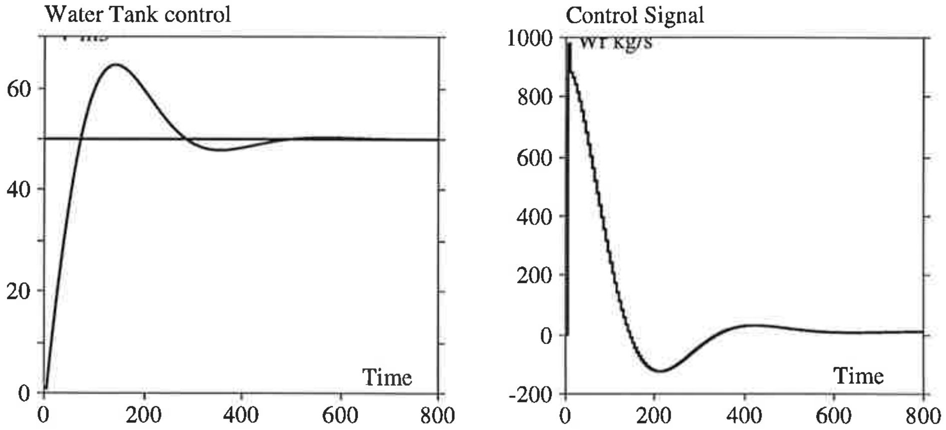

Figure 5.15: Water Tank Control and *Make-Up Water* Flow

Chapter 6

Conclusions

A modified version of a gas turbine unit which belongs to the class of evaporative gas turbine cycles is presented and modeled. The Humid Air Gas Turbine Cycle, **HAT cycle** is implemented in the Omola language and simulated in the OmSim environment, which are developed at The Department of Automatic Control, Lund Institute of Technology. The cycle consists of several process units and subunits that are modeled separately for the purpose of reuseability.

The major units in the modeled cycle are a humidification tower, a recuperator, an air-cooled expander, and a water circulation system. The cycle is simulated using models of fluid compression, expansion, mass and heat transfer, mass and energy balances, and heat generation.

An analysis of the non-linearities of the discretecized humidification tower results in a proposed stabilization of the tower via a state-feedback control law. Thereby, the transient thermal characteristics of the discretecized tower is controlled by pole-placement.

The simulated results are compared to operational data from different power plants by thermodynamic performance criteria such as energy utilization factor, fuel energy saving ratio, and artificial thermal efficiency. Variations in these criteria is presented as a function of different process parameters.

Chapter 7

References

7.1 The Bibliography

Bibliography

[1] B.Nilsson : *Object-Oriented Modeling of Chemical Processes.* Dept. Automatic Control, Lund Institute of Technology, August 1993.

[2] B.Wittenmark, K.J.Åstöm, and S.B. Jörgensen :*Process Control.* Dept. Automatic Control, Lund Institute of Technology, 1991.

[3] B.Wittenmark, K.J.Åstöm: *Computer-Controlled Systems, Theory and Design.* Prentice Hall, International Editions, 1990.

[4] H.K. Khalil:*Nonlinear Systems.* MacMillian, 1992.

[5] P.M. Rosen : *Evaporative Gas Turbine Cycles.* Dept. of Heat and Power Engineering, Lund Institute of Technology, March 1993.

[6] B. Nilsson : *Structured Modelling of Chemical Processes. An Object-Oriented Approach.* Lic Tech thesis TFRT-3203. Dept. Automatic Control, Lund Institute of Technology, 1989.

[7] W.L. Luyben : *Process Modelling, Simulation, and Control for Chemical Engineers.* McGraw-Hill, Second Edition,1993.

[8] M. Andersson : *Object-Oriented Modelling and Simulation of Hybrid Systems.* PhD Thesis. Dept. Automatic Control, Lund Institute of Technology, December 1994.

[9] J.H. Horloch : *CoGeneration: Combined Heat and Power, Thermodynamics and Economics.* Pergamon Press, 1987.

[10] A.R.J. Timmermans : *Combined Cycles and their Possibilities. Lecture Series, Combined Cycles for Power Generation.* Von Karman Institute for Fluid Mechanics, Rhode Saint Genese, Belgium, 1978.

[11] R.W. Haywood : *Analysis of Engineering Cycles.*3rd Edition. Pergamon Press, Oxford, 1980.

[12] K. Bammert : *Twenty-Five Years of Operating Experience with Coal-Fired Closed Cycle Gas Turbine Cogeneration Plant at Coburg.* Trans. ASME Journal of Engineering for Power, 105, 806-815, 1983.

[13] J. Flad :*Operation of a Combined District Heating and Power Production Scheme with a Combined Gas Turbine.*Lecture Series. Von Karman Institute for Fluid Mechanics, Rhode Saint Genese, Belgium, 1978.

[14] R.P. Allen and J.M. Kovacik :*Gas Turbines: Cogeneration Principles and Practice.* Trans. ASME Journal of Engineering for Gas Turbines and Power, 106, 4,725-730, 1984.

[15] S.E. Matsson and M. Andersson :*The Ideas Behind Omola.* IEEE Symposium on Computer-Aided Control System Design, March 17-19, 1992, Napa, California.

[16] M. Andersson :*Omsim and Omola Tutorial and User's Manual.* Version 3.4 .Dept. Automatic Control, Lund Institute of Technology, March 1995.

[17] M. Andersson, D. Brück, S.E. Matsson, and T. Schöntal : *Omsim-An Integrated Interactive Environment for Object-Oriented Modeling and Simulation.* Dept. Automatic Control,Lund Institute of Technology, Lund.

[18] Å. Jernqvist : *Mass, Heat, and Impulse Transport- Fundamental Theories.* Dept. Chemical Engineering. Lund Institute of Technology.

[19] Å. Jernqvist : *Air and Water Systems.* Dept. Chemical Engineering. Lund Institute of Technology.

[20] G. *Heyen*[2], K. *Murphy*[2], D. *Marchio*[2], P.*Kalata*[3], B. *Kalitventzeff*[1], and E. *Marechal*[1] :*Dynamic Simulation and Control of Gas Turbines and Compressor Systems.* 1/LASSC, Universite de Liege, Sart Tilman B6, B4000 Liege, Belgium. 2/Centre d'Energetique, Ecole des Mines de Paris, Paris, France. 3/Electrical and Computer Engineering Dept, Drexel University, PA, U.S.A. Computers chem. Engng Vol. 18, No.11/12, pp. 1071-1082, February 1994.

[21] T.D.Eastop and A.McConkey:*Applied Thermodynamics for Engineering Technologists.* Fifth Edition.Pergamon 1993.

[22] A. Hussain :*Development of Simplified Gas Turbine Simulation Models.* Msc Dissertation, University of Manchesters Institute of Technology, 1988.

[23] A.*Hussain*[1] and H.*Seifi*[2]: *Dynamic Modeling of a Single Shaft Gas Turbine.* 1/Universiti Kebangson, Malaysia. 2/University of Tarbiat-Modarres, Tehran, Iran. IFAC Control of Power Plants and Power Systems, Munich, Germany, 1992.

[24] G. *Heyen*[2], B. *Kalitventzeff*[1], P.*Hutchinson*[2], Hoi *Yeung*[2], M.*Gill*[3]:*Simulation of Fast Transients in Fluid Transport Equipments and Utility Networks.*1/LASSC, Universite de Liege, Sart Tilman B6, B4000 Liege, Belgium.2/School of Mechanical Engineering, Cranfield Institute of Technology, Bedford MK43 OAL(UK). 3/W.S Atkins, Woodcote Grove, Ashley Road, Epsom Surrey KT18 5BW(UK). European Symposium on Computer Aided Process Engineering-1,s110,1987.

[25] J.M.Coulson and J.F.Richardson:*Chemical Engineering.*Volume 1 and 2.Fourth Edition.Pergamon 1993.

Appendix A

The Program Codes

A.1 Omola Code: The Hat Cycle

```
LIBRARY HatCycle6;

% This is the copy of the HatMod2.
% A Classic Power Plant With a Gas Turbine &Comp. and a CC
% with tower and external water .Two comp exists here.
% Enourmous diff. with differential HeatEx
% Cooling System is incorporated;Dynamic.
% HeatEx is static.
% Copied from HatMod232.om on May 2.
% A model of Split also implemented.
% A try with events shall be carried out.
% Copied from the HatCycle4.om on 14 juni.
%Shall try to incorporate dynamic HeatEx.2 try.


InCon ISA RecordTerminal WITH
% The connecting Terminals between the Models.
% These are used on the In Connection.

  Pin ISA SimpleInput;
  Win ISA SimpleInput;
  Hin ISA SimpleInput;

END;

OutCon ISA RecordTerminal WITH
% The connecting terminals.
% These are used for the out going.

  Pout ISA SimpleOutput;
  Wout ISA SimpleOutput;
  Hout ISA SimpleOutput;
```

```
END;


FilmSys ISA Base::Model WITH
% The model used to describe the Film Theory

terminals:
Qing      ISA SimpleTerminal;
Qoutg     ISA SimpleTerminal;
Qoutl     ISA SimpleTerminal;
Houtg     ISA SimpleTerminal;
Houtl     ISA SimpleTerminal;
mg        ISA SimpleTerminal;
ml        ISA SimpleTerminal;

parameters:

AlphaG ISA Parameter WITH default:=150; END; %This represents alpha*A
Hevap  ISA Parameter WITH default:=3500;END; % The heat of evaporation
Cpl    ISA Parameter WITH default:=4.1; END; % The heat capacity of water
Cpg    ISA Parameter WITH default:=1.8; END; % The heat capacity of air

equations:
% Consider if Hevap=Hout during simulation.

mg       = Qoutg/ (Hevap-(Houtl+Houtg));
Qoutg    = AlphaG*(Houtg/Cpg-Houtl/Cpl);
Qing     = mg*(Hevap-Houtg);
mg       = ml ;

END;


AirSys ISA Model WITH
% A model of the Air Side for
% the decomposed Cut

Terminals:

% model terminals to other units
AirSysIn  ISA InCon;
AirSysOut ISA OutCon;

% model connection to the film membrane
Houtg    ISA SimpleTerminal;
Qout     ISA SimpleTerminal;
Qin      ISA SimpleTerminal;
mg       ISA SimpleTerminal;
```

```
Parameters:

Cpg       ISA Parameter WITH default:=1.8;END;% The heat capacity of air
m         ISA Parameter WITH default:=40 ;END;% The mass of air
Fric      ISA Parameter WITH default:=1.0;END;% Friction coeff on the airSide


Variables:

Tin     ISA Variable;
Q       ISA Variable;

Equations:

AirSysOut.Wout  = AirSysIn.Win + mg ;
Tin            := AirSysIn.Hin / Cpg ;
AirSysOut.Hout' = 1/m*(AirSysIn.Win*(AirSysIn.Hin-AirSysOut.Hout) + Q - mg*AirSysOut.Ho
Houtg           = AirSysOut.Hout;
Q               =  -Qout + Qin ;
AirSysOut.Pout := AirSysIn.Pin - Fric*SQR(AirSysOut.Wout);

END;


WatSys ISA Base::Model WITH

terminals:

% model connection to the outside units
WatSysIn  ISA InCon;
WatSysOut ISA OutCon;

%model connection to the film memebrane
Houtl     ISA SimpleTerminal;
ml        ISA SimpleTerminal;
Qout      ISA SimpleTerminal;

Parameters:

Cpl       ISA Parameter WITH default:=4.1;END;% heat capacity of water
m         ISA Parameter WITH default:=20;END;% the mass of water
Fric      ISA Parameter WITH default:=0.1;END;%Friction Coeff on the wat side


Equations:

WatSysOut.Wout  =   WatSysIn.Win - ml;
WatSysOut.Hout' =   1/m*(WatSysIn.Hin-WatSysOut.Hout)*WatSysIn.Win;
Qout            =   ml*Houtl;
Houtl           =   WatSysOut.Hout;
```

```
WatSysOut.Pout :=   WatSysIn.Pin - Fric*SQR(WatSysOut.Wout);

END;


AirWat ISA Base::Model WITH

submodels:
% Submodels composing the Air and Water System
Air  ISA AirSys;
Film ISA FilmSys;
Wat  ISA WatSys;

terminals:
% terminals of the Tray model to the outside
AirIn  ISA InCon;
WatIn  ISA InCon;
AirOut ISA OutCon;
WatOut ISA OutCon;

connections:

% The connections from the Air System
Air.Houtg AT Film.Houtg;
Air.Qout  AT Film.Qoutg;
Air.Qin   AT Film.Qing;
Air.mg    AT Film.mg;

% The connections from the Water System
Wat.Houtl AT Film.Houtl;
Wat.Qout  AT Film.Qoutl;
Wat.ml    AT Film.ml;

Equation:

Air.AirSysIn.Pin:=AirIn.Pin;
Air.AirSysIn.Hin:=AirIn.Hin;
Air.AirSysIn.Win:=AirIn.Win;
Wat.WatSysIn.Pin:=WatIn.Pin;
Wat.WatSysIn.Hin:=WatIn.Hin;
Wat.WatSysIn.Win:=WatIn.Win;

AirOut.Pout := Air.AirSysOut.Pout;
AirOut.Wout := Air.AirSysOut.Wout;
AirOut.Hout := Air.AirSysOut.Hout;
WatOut.Pout := Wat.WatSysOut.Pout;
WatOut.Wout := Wat.WatSysOut.Wout;
WatOut.Hout := Wat.WatSysOut.Hout;

END;
```

```
TowerModel ISA Base::Model WITH
% Five tray modell of the real thing.

submodels:
% Submodels of the Tray Model composed
Tray1 ISA AirWat;
Tray2 ISA AirWat;
Tray3 ISA AirWat;
Tray4 ISA AirWat;
Tray5 ISA AirWat;

terminals:
% THe terminmal
TowerAirIn  ISA InCon;
TowerAirOut ISA OutCon;
TowerWatIn  ISA InCon;
TowerWatOut ISA OutCon;

Variables:
dTin,dTout ISA Variable;% Pinch studies over the AbsTower

parameters:

connections:
Tray1.AirIn AT TowerAirIn;
TowerAirOut AT Tray5.AirOut;

Tray2.AirIn AT Tray1.AirOut;
Tray3.AirIn AT Tray2.AirOut;
Tray4.AirIn AT Tray3.AirOut;
Tray5.AirIn At Tray4.AirOut;

Tray1.WatIn AT Tray2.WatOut;
Tray2.WatIn AT Tray3.WatOut;
Tray3.WatIn AT Tray4.WatOut;
Tray4.WatIn AT Tray5.WatOut;
Tray5.WatIn AT TowerWatIn;
TowerWatOut AT Tray1.WatOut;

Equations:
dTin:=(Tray5.AirOut.Hout-Tray5.WatIn.Hin);
dTout:=(Tray1.AirIn.Hin-Tray1.WatOut.Hout);

END;


Tank ISA Model WITH
```

```
Terminals:
% The terminals or the abstract surface of the Tank System
InFlow  ISA InCon;
OutFlow ISA OutCon;

Parameter:

   Vref     ISA Parameter WITH default:=50  ;END;% The volume ref. value
   Ro       ISA Parameter WITH default:=1000;END;% The density of water
   Hr       ISA Parameter WITH default:=100 ;END;% Water Enthalpy
   PressEq  ISA Parameter WITH default:=12E5;END;% Pressure Equaliser
   MassFlow ISA Parameter WITH default:=100 ;END;% To the intercooler
   k        ISA Parameter WITH default:=10  ;END;% The prop. controller
   A        ISA Parameter WITH default:=50  ;END;%tank area
   h        ISA Parameter WITH default:=0.5 ;END;% The samplings period
   ti       ISA Parameter WITH default:=10  ;END;% Integrator Time Constant

Variables:

m ISA Variable WITH initial:=1000;END;
V ISA Variable WITH initial:=50  ;END;

P ISA Variable; % Just to test the pressure loss

Equations:

   OutFlow.Pout  := PressEq;
   P             := InFlow.Pin;
   OutFlow.Wout  := MassFlow;
   V             := m/Ro;
   m'             = InFlow.Win + Wr - OutFlow.Wout;
   m*OutFlow.Hout'= InFlow.Win*(InFlow.Hin-OutFlow.Hout)+Wr*(Hr-OutFlow.Hout);

Events:
inte, e, u ,Wr TYPE DISCRETE REAL;
  Init, Sample ISAN Event;
  ONEVENT Init OR Sample DO
    new(e) := Vref - V;
    new(inte) := inte + k*e*h/ti;
    new(u) := k*new(e) + inte;
    SCHEDULE(Sample,h);
  END;
  Wr:= u;




END;

%-------------------
```

```
HeatEx2 ISA Model WITH
Parameters:
Kc  ISA Parameter;
Kw  ISA Parameter;
U   ISA Parameter;
A   ISA Parameter;
m   ISA Parameter WITH default:=2000;END;
Cpg ISA Parameter WITH default:=1.4;END;

Terminals:

HeatExWarmSideIn  ISA InCon;
HeatExWarmSideOut ISA OutCon;
HeatExColdSideIn  ISA InCon;
HeatExColdSideOut ISA OutCon;

Variables:

Diff1   ISA Variable;
Diff2   ISA Variable;
Theta1  ISA Variable WITH initial:=5000;END;% Pinch on the Warm Side
Theta2  ISA Variable WITH initial:=4000;END;% Pinch on the Cold Side
Q       ISA Variable ;

E1 ISA Variable;
E2 ISA Variable;
E3 ISA Variable;
E4 ISA Variable;


Equations:
HeatExWarmSideOut.Pout:=HeatExWarmSideIn.Pin-Kw*SQR(HeatExWarmSideOut.Wout);
HeatExWarmSideOut.Wout:=HeatExWarmSideIn.Win;
HeatExColdSideOut.Pout:=HeatExColdSideIn.Pin-Kc*SQR(HeatExColdSideOut.Wout);
HeatExColdSideOut.Wout:=HeatExColdSideIn.Win;

E1=HeatExColdSideIn.Win*HeatExColdSideIn.Hin;
E2=HeatExColdSideOut.Wout*HeatExColdSideOut.Hout;
E3=HeatExWarmSideIn.Win*HeatExWarmSideIn.Hin;
E4=HeatExWarmSideOut.Wout*HeatExWarmSideOut.Hout;

Diff1 =(E1-E2);
Diff2 =(E3-E4);

m*HeatExColdSideOut.Hout'=Diff1+Q;
m*HeatExWarmSideOut.Hout'=Diff2-Q;
```

```
Theta1=HeatExWarmSideIn.Hin-HeatExColdSideOut.Hout;
Theta2=HeatExWarmSideOut.Hout-HeatExColdSideIn.Hin;

Q = U*A/Cpg*(Theta1-Theta2)/LN(abs(Theta1/Theta2));



END;
%-------------------

HeatEx ISA Model WITH
Parameters:
Kc ISA Parameter;
Kw ISA Parameter;

Terminals:
HeatExWarmSideIn  ISA InCon;
HeatExWarmSideOut ISA OutCon;
HeatExColdSideIn  ISA InCon;
HeatExColdSideOut ISA OutCon;

Variables:
Pinch1 ISA Variable;
Pinch2 ISA Variable;
E1     ISA Variable;
E2     ISA Variable;
E3     ISA Variable;
E4     ISA Variable;

Equations:
HeatExWarmSideOut.Pout:=HeatExWarmSideIn.Pin-Kw*SQR(HeatExWarmSideOut.Wout);
HeatExWarmSideOut.Wout:=HeatExWarmSideIn.Win;
HeatExColdSideOut.Pout:=HeatExColdSideIn.Pin-Kc*SQR(HeatExColdSideOut.Wout);
HeatExColdSideOut.Wout:=HeatExColdSideIn.Win;

E1=HeatExColdSideIn.Win*HeatExColdSideIn.Hin;
E2=HeatExColdSideOut.Wout*HeatExColdSideOut.Hout;
E3=HeatExWarmSideIn.Win*HeatExWarmSideIn.Hin;
E4=HeatExWarmSideOut.Wout*HeatExWarmSideOut.Hout;

Pinch1 =(E2-E1);
Pinch2 =(E3-E4);

Pinch1- Pinch2 =0;

END;
```

```
Generator  ISA  Model WITH

Terminals:
PowerIn ISA SimpleTerminal;

Variable:
P ISA Variable;

Equation:
P:=PowerIn;

END;


WorkUnit ISA Model WITH

Parameters:
GasCnst   ISA Parameter WITH default:=8.314;END;
MolMass   ISA Parameter WITH default:=28;END;
Xi        ISA Parameter WITH default:=1.4;END;
Cpg       ISA Parameter WITH default:=1.4;END;
Nuo       ISA Parameter WITH default:=0.9;END;


Terminals:
UnitConIn  ISA InCon;
UnitConOut ISA OutCon;


Variables:
ra,Work,k ISA Variable;


Equations:

ra               := UnitConOut.Pout/UnitConIn.Pin;
k                :=  (Xi/(1-Xi))/Nuo;
UnitConOut.Wout:= UnitConIn.Win;

Work  = k*UnitConIn.Win*GasCnst*UnitConIn.Hin/(Molmass*Cpg)*(ra^(-1/k)-1);
UnitConOut.Hout = UnitConIn.Hin*ra^(-1/k);


END;



Combust ISA Model WITH
```

```
Parameters:
K ISA Parameter WITH default:=0.1;END;% Friction coefficient

Terminals:
UnitConIn  ISA InCon;
UnitConOut ISA OutCon;

FeulFlow    ISA SimpleTerminal;
FeulEnergy  ISA SimpleTerminal;

Equations:

UnitConIn.Win + FeulFlow = UnitConOut.Wout;
UnitConIn.Win*UnitConIn.Hin + FeulFlow*FeulEnergy = UnitConOut.Hout*UnitConOut.Wout;
UnitConOut.Pout:= UnitConIn.Pin-K*SQR(UnitConIn.Win);

END;




SplitModel ISA Model WITH

Parameters:
X         ISA Parameter WITH default:=0.05;END;% Procentage by-pass flow

Terminals:
InFlow      ISA InCon;
OutFlow     ISA OutCon;
CoolAirFlow ISA SimpleOutput;
CoolAirEnth ISA SimpleOutput;

Equations:
CoolAirFlow:=X*InFlow.Win;
CoolAirEnth:=InFlow.Hin;

OutFlow.Hout:=InFlow.Hin;
OutFlow.Wout:=(1-X)*InFlow.Win;
OutFlow.Pout:=InFlow.Pin;

END;


Turbine     ISA Model WITH

Parameters:
GasCnst   ISA Parameter WITH default:=8.314;END;
```

```
MolMass    ISA Parameter WITH default:=28;END;
Xi         ISA Parameter WITH default:=1.4;END;
Cpg        ISA Parameter WITH default:=1.4;END;
m          ISA Parameter WITH default:=5;END;
Nuo        ISA Parameter WITH default:=0.9;END;

Terminals:
UnitConIn   ISA InCon;
UnitConOut  ISA OutCon;
CoolAirFlow ISA SimpleInput;
CoolAirEnth ISA SimpleTerminal;

Variables:
ra,Work,k    ISA Variable;
EnthTurbine  ISA Variable WITH initial:=1000;END;
FlowTurbine  ISA Variable WITH initial:=10;END;

Equations:

ra               := UnitConOut.Pout/UnitConIn.Pin;
k                :=  (Xi/(1-Xi))/Nuo;
UnitConOut.Wout:=FlowTurbine;

Work  := k*FlowTurbine*GasCnst*EnthTurbine/(Molmass*Cpg)*(ra^(-1/k)-1);
UnitConOut.Hout := EnthTurbine*ra^(-1/k);
FlowTurbine     :=  UnitConIn.Win + CoolAirFlow;

m*EnthTurbine'= UnitConIn.Win*(UnitConIn.Hin-EnthTurbine) + CoolAirFlow*(CoolAirEnth-Ent

END;

Comp        ISA WorkUnit;

Recuporator ISA HeatEx2 WITH

Parameter:

Kc  ISA Parameter WITH default:=1;END;
Kw  ISA Parameter WITH default:=1;END;
U   ISA Parameter WITH default:=0.05;END;
A   ISA Parameter WITH default:=1000;END;

END;

InterCooler ISA HeatEx WITH

Parameters:
Kc  ISA Parameter WITH default:=1;END;
Kw  ISA Parameter WITH default:=1;END;
```

```
END;


PowerPlant  ISA Model WITH

submodels:
GT      ISA Turbine;
HC      ISA Comp;
LC      ISA Comp;
CC      ISA Combust;
IC      ISA InterCooler;
Gen     ISA Generator;
Rec     ISA Recuporator;
Tower   ISA TowerModel;
Eq      ISA Tank;
Split   ISA SplitModel;
TempCon ISA TempConModel;


parameters:

AirFlow        ISA Parameter WITH default:=30;END;
AirPress       ISA Parameter WITH default:=1E5;END;
LCPressOut     ISA Parameter WITH default:=2.64E5;END;
HCPressOut     ISA Parameter WITH default:=7E5;END;
EntalpiIn      ISA Parameter WITH default:=85;END;
Feul           ISA Parameter WITH default:=1.2;END;
Heat           ISA Parameter WITH default:=43E3;END;
ICWarmSideOut  ISA Parameter WITH default:=100;END;

connections:

GT.UnitConIn            AT CC.UnitConOut;
HC.UnitConOut           AT Tower.TowerAirIn;
Rec.HeatExColdSideIn    AT Split.OutFlow;
Tower.TowerAirOut       AT Split.InFlow;
GT.UnitConOut           AT Rec.HeatExWarmSideIn;
CC.UnitConIn            AT Rec.HeatExColdSideOut;
LC.UnitConOut           AT IC.HeatExWarmSideIn;
IC.HeatExWarmSideOut    AT HC.UnitConIn;
Tower.TowerWatIn        AT IC.HeatExColdSideOut;
Eq.InFlow               AT Tower.TowerWatOut;
IC.HeatExColdSideIn     AT Eq.OutFlow;
Split.CoolAirFlow       AT GT.CoolAirFlow;
Split.CoolAirEnth       AT GT.CoolAirEnth;
TempCon.X               AT Split.X;


Equations:
Gen.PowerIn:=GT.Work-LC.Work-HC.Work;
```

```
TempCon.H1:=CC.UnitConOut.Hout;
TempCon.H2:=GT.EnthTurbine;


LC.UnitConIn.Hin:=EntalpiIn;
LC.UnitConIn.Pin:=AirPress;
LC.UnitConIn.Win:=AirFlow;

HC.UnitConOut.Pout:=HCPressOut;

LC.UnitConOut.Pout:=LCPressOut;


GT.UnitConOut.Pout:=1E5;

CC.FeulFlow:=Feul;
CC.FeulEnergy:=Heat;
IC.HeatExWarmSideOut.Hout:=ICWarmSideOut;


Variables:
Lambda ISA Variable;
Eta    ISA Variable;
ART    ISA Variable;
EUF    ISA Variable;
FESR   ISA Variable;
IHR    ISA Variable;
Equations:


Eta    = ABS(Gen.P)/(CC.FeulFlow*CC.FeulEnergy);
Lambda = ABS(Rec.Q)/ABS(Gen.P);
EUF    = (ABS(Gen.P)+ABS(Rec.Q))/(CC.FeulFlow*CC.FeulEnergy);
ART    = Eta /(1-(Rec.Q/(0.90*CC.FeulFlow*CC.FeulEnergy)));
FESR   = 1-((0.40/Eta)/(1+(Lambda*0.40/0.90)));
IHR    =1/eta-Lambda/0.90;

END;
```

A.2 Simnon Code: The Stabilized Tray Model

```
CONTINUOUS SYSTEM trmod

" The Tray model

INPUT u1 u2
OUTPUT y1 y2
```

```
STATE x1 x2
DER dx1 dx2




dx1 =1/mas* (-asin*x1+asin*u1-cnst*x1*x1+cnst*x1*x2)
dx2 =1/mws* (-wsin*x2+wsin*u2)

y1=x1
y2=x2

cnst=2*AlphaG/(Hevap*Cpg)
AlphaG:250 "Heat transfer coefficient
Hevap:4500 "Enthalpy of evaporation
Cpg:1.4     "Air heat capacity
Cpl:4.1     "Water heat capacity
asin:40     "Air Flow into the tray
wsin:70    "Water Flow into the tray
mas:50      "accumulated mass air side
mws:20      "accumulated mass water side



END


DISCRETE SYSTEM sfb
"
INPUT y1 y2
OUTPUT u1 u2
TIME t
TSAMP ts

w1=cnst/asin*((y1*y1/Cpg)-(y1*y2/Cpl))+v
v=k1*y1+k2*y2
w2=200 + 100*SIGN(sin(omg*t))
u1=w1
u2=w2

cnst=2*AlphaG/(Hevap*Cpg)
ts=t+h

h:0.01      "sampling time
k1:0.6    "
k2:1.0        "
omg:0.15

AlphaG:250 "Heat transfer coefficient
Hevap:4500 "Enthalpy of evaporation
```

```
Cpg:1.4     "Air heat capacity
Cpl:4.1     "Water heat capacity
asin:40     "Air Flow into the tray
wsin:70    "Water Flow into the tray
mas:50      "accumulated mass air side
mws:20      "accumulated mass water side



END


CONNECTING SYSTEM conn
"
y1[sfb]=y1[trmod]
y2[sfb]=y2[trmod]

u1[trmod]=u1[sfb]
u2[trmod]=u2[sfb]



END


MACRO mcsim
SYST trmod sfb conn
HCOPY ON
STORE u1[sfb] u2[sfb] y1[trmod] y2[trmod] v[sfb]
PAR wsin[sfb]:70
PAR wsin[trmod]:70
SIMU 0 80
SPLIT 2 1
AXES H 0 90 V 0 400
AXES H 0 90 V -100 900
AREA 1 1
SHOW u2
TEXT'                                        U2'
AREA 2 1
SHOW v y1-MARK
TEXT'                                     V and Y1         '

MARK A 0 5.5
MARK " (kJ/kg)

MARK A 18 1.5
MARK " Time

MARK A 0 12.5
MARK " (kJ/kg)
```

```
MARK A 18 8.5
MARK " Time
HCOPY META/TRAYSFB

END
```

A.3 Snapshot of the HatCycle6 Model

```
% Snapshot of Simulator 1 dated Wed Jun 21 13:24:38 1995
% PowerPlant ISA Model;


% Parameters:
PowerPlant.GT.GasCnst.default := 8.314;
PowerPlant.GT.MolMass.default := 28.0;
PowerPlant.GT.Xi.default := 1.4;
PowerPlant.GT.Cpg.default := 1.4;
PowerPlant.GT.m.default := 5.0;
PowerPlant.GT.Nuo.default := 0.9;
PowerPlant.HC.GasCnst.default := 8.314;
PowerPlant.HC.MolMass.default := 28.0;
PowerPlant.HC.Xi.default := 1.4;
PowerPlant.HC.Cpg.default := 1.4;
PowerPlant.HC.Nuo.default := 0.9;
PowerPlant.LC.GasCnst.default := 8.314;
PowerPlant.LC.MolMass.default := 28.0;
PowerPlant.LC.Xi.default := 1.4;
PowerPlant.LC.Cpg.default := 1.4;
PowerPlant.LC.Nuo.default := 0.9;
PowerPlant.CC.k.default := 0.1;
PowerPlant.IC.Kc.default := 1.0;
PowerPlant.IC.Kw.default := 1.0;
PowerPlant.Rec.Kc.default := 1.0;
PowerPlant.Rec.Kw.default := 1.0;
PowerPlant.Rec.U.default := 0.05;
PowerPlant.Rec.A.default := 1000.0;
PowerPlant.Rec.m.default := 2000.0;
PowerPlant.Rec.Cpg.default := 1.4;
PowerPlant.Tower.Tray1.Air.Cpg.default := 1.8;
PowerPlant.Tower.Tray1.Air.m.default := 40.0;
PowerPlant.Tower.Tray1.Air.Fric.default := 1.0;
PowerPlant.Tower.Tray1.Film.AlphaG.default := 150.0;
PowerPlant.Tower.Tray1.Film.Hevap.default := 3500.0;
PowerPlant.Tower.Tray1.Film.Cpl.default := 4.1;
PowerPlant.Tower.Tray1.Film.Cpg.default := 1.8;
PowerPlant.Tower.Tray1.Wat.Cpl.default := 4.1;
PowerPlant.Tower.Tray1.Wat.m.default := 20.0;
```

```
PowerPlant.Tower.Tray1.Wat.Fric.default := 0.1;
PowerPlant.Tower.Tray2.Air.Cpg.default := 1.8;
PowerPlant.Tower.Tray2.Air.m.default := 40.0;
PowerPlant.Tower.Tray2.Air.Fric.default := 1.0;
PowerPlant.Tower.Tray2.Film.AlphaG.default := 150.0;
PowerPlant.Tower.Tray2.Film.Hevap.default := 3500.0;
PowerPlant.Tower.Tray2.Film.Cpl.default := 4.1;
PowerPlant.Tower.Tray2.Film.Cpg.default := 1.8;
PowerPlant.Tower.Tray2.Wat.Cpl.default := 4.1;
PowerPlant.Tower.Tray2.Wat.m.default := 20.0;
PowerPlant.Tower.Tray2.Wat.Fric.default := 0.1;
PowerPlant.Tower.Tray3.Air.Cpg.default := 1.8;
PowerPlant.Tower.Tray3.Air.m.default := 40.0;
PowerPlant.Tower.Tray3.Air.Fric.default := 1.0;
PowerPlant.Tower.Tray3.Film.AlphaG.default := 150.0;
PowerPlant.Tower.Tray3.Film.Hevap.default := 3500.0;
PowerPlant.Tower.Tray3.Film.Cpl.default := 4.1;
PowerPlant.Tower.Tray3.Film.Cpg.default := 1.8;
PowerPlant.Tower.Tray3.Wat.Cpl.default := 4.1;
PowerPlant.Tower.Tray3.Wat.m.default := 20.0;
PowerPlant.Tower.Tray3.Wat.Fric.default := 0.1;
PowerPlant.Tower.Tray4.Air.Cpg.default := 1.8;
PowerPlant.Tower.Tray4.Air.m.default := 40.0;
PowerPlant.Tower.Tray4.Air.Fric.default := 1.0;
PowerPlant.Tower.Tray4.Film.AlphaG.default := 150.0;
PowerPlant.Tower.Tray4.Film.Hevap.default := 3500.0;
PowerPlant.Tower.Tray4.Film.Cpl.default := 4.1;
PowerPlant.Tower.Tray4.Film.Cpg.default := 1.8;
PowerPlant.Tower.Tray4.Wat.Cpl.default := 4.1;
PowerPlant.Tower.Tray4.Wat.m.default := 20.0;
PowerPlant.Tower.Tray4.Wat.Fric.default := 0.1;
PowerPlant.Tower.Tray5.Air.Cpg.default := 1.8;
PowerPlant.Tower.Tray5.Air.m.default := 40.0;
PowerPlant.Tower.Tray5.Air.Fric.default := 1.0;
PowerPlant.Tower.Tray5.Film.AlphaG.default := 150.0;
PowerPlant.Tower.Tray5.Film.Hevap.default := 3500.0;
PowerPlant.Tower.Tray5.Film.Cpl.default := 4.1;
PowerPlant.Tower.Tray5.Film.Cpg.default := 1.8;
PowerPlant.Tower.Tray5.Wat.Cpl.default := 4.1;
PowerPlant.Tower.Tray5.Wat.m.default := 20.0;
PowerPlant.Tower.Tray5.Wat.Fric.default := 0.1;
PowerPlant.Eq.Vref.default := 50.0;
PowerPlant.Eq.Ro.default := 1000.0;
PowerPlant.Eq.Hr.default := 85.0;
PowerPlant.Eq.tsamp.default := 0.5;
PowerPlant.Eq.PressEq.default := 1.2e+06;
PowerPlant.Eq.MassFlow.default := 100.0;
PowerPlant.Eq.k.default := 10.0;
PowerPlant.Eq.A.default := 50.0;
PowerPlant.TempCon.h.default := 5.0;
```

```
PowerPlant.TempCon.Href.default := 2000.0;
PowerPlant.AirFlow.default := 30.0;
PowerPlant.AirPress.default := 100000.0;
PowerPlant.LCPressOut.default := 264000.0;
PowerPlant.HCPressOut.default := 700000.0;
PowerPlant.EntalpiIn.default := 85.0;
PowerPlant.Feul.default := 1.2;
PowerPlant.Heat.default := 43000.0;
PowerPlant.ICWarmSideOut.default := 100.0;

% Continuous State Variables:
PowerPlant.GT.EnthTurbine.initial := 1918.85;
PowerPlant.CC.UnitConOut.Hout.initial := 1918.36;
PowerPlant.IC.HeatExColdSideOut.Hout.initial := 76.4415;
PowerPlant.Rec.HeatExWarmSideOut.Hout.initial := 519.905;
PowerPlant.Rec.HeatExColdSideOut.Hout.initial := 709.233;
PowerPlant.Tower.Tray1.Air.AirSysOut.Hout.initial := 124.729;
PowerPlant.Tower.Tray1.Wat.WatSysOut.Hout.initial := 75.6861;
PowerPlant.Tower.Tray2.Air.AirSysOut.Hout.initial := 121.404;
PowerPlant.Tower.Tray2.Wat.WatSysOut.Hout.initial := 75.8528;
PowerPlant.Tower.Tray3.Air.AirSysOut.Hout.initial := 118.493;
PowerPlant.Tower.Tray3.Wat.WatSysOut.Hout.initial := 76.0109;
PowerPlant.Tower.Tray4.Air.AirSysOut.Hout.initial := 115.891;
PowerPlant.Tower.Tray4.Wat.WatSysOut.Hout.initial := 76.1613;
PowerPlant.Tower.Tray5.Air.AirSysOut.Hout.initial := 113.52;
PowerPlant.Tower.Tray5.Wat.WatSysOut.Hout.initial := 76.3046;
PowerPlant.Eq.OutFlow.Hout.initial := 73.711;
PowerPlant.Eq.m.initial := 7478.84;

% Auxiliary Variables:
PowerPlant.GT.UnitConIn.Pin.initial := 691425.0;
PowerPlant.GT.UnitConIn.Win.initial := 41.9716;
PowerPlant.GT.UnitConIn.Hin.initial := 1918.36;
PowerPlant.GT.UnitConOut.Pout.initial := 100000.0;
PowerPlant.GT.UnitConOut.Wout.initial := 41.9716;
PowerPlant.GT.UnitConOut.Hout.initial := 1167.1;
PowerPlant.GT.CoolAirFlow.initial := 0.0;
PowerPlant.GT.CoolAirEnth.initial := 113.52;
PowerPlant.GT.ra.initial := 0.144629;
PowerPlant.GT.ra.initial.initial := 0.0;
PowerPlant.GT.Work.initial := 26024.4;
PowerPlant.GT.Work.initial.initial := 0.0;
PowerPlant.GT.k.initial := -3.88889;
PowerPlant.GT.k.initial.initial := 0.0;
PowerPlant.GT.EnthTurbine.initial.initial := 1000.0;
PowerPlant.GT.FlowTurbine.initial := 41.9716;
PowerPlant.GT.FlowTurbine.initial.initial := 10.0;
PowerPlant.HC.UnitConIn.Pin.initial := 263100.0;
PowerPlant.HC.UnitConIn.Win.initial := 30.0;
PowerPlant.HC.UnitConIn.Hin.initial := 100.0;
```

```
PowerPlant.HC.UnitConOut.Pout.initial := 700000.0;
PowerPlant.HC.UnitConOut.Wout.initial := 30.0;
PowerPlant.HC.UnitConOut.Hout.initial := 128.612;
PowerPlant.HC.ra.initial := 2.66059;
PowerPlant.HC.ra.initial.initial := 0.0;
PowerPlant.HC.Work.initial := -707.965;
PowerPlant.HC.Work.initial.initial := 0.0;
PowerPlant.HC.k.initial := -3.88889;
PowerPlant.HC.k.initial.initial := 0.0;
PowerPlant.LC.UnitConIn.Pin.initial := 100000.0;
PowerPlant.LC.UnitConIn.Win.initial := 30.0;
PowerPlant.LC.UnitConIn.Hin.initial := 85.0;
PowerPlant.LC.UnitConOut.Pout.initial := 264000.0;
PowerPlant.LC.UnitConOut.Wout.initial := 30.0;
PowerPlant.LC.UnitConOut.Hout.initial := 109.102;
PowerPlant.LC.ra.initial := 2.64;
PowerPlant.LC.ra.initial.initial := 0.0;
PowerPlant.LC.Work.initial := -596.373;
PowerPlant.LC.Work.initial.initial := 0.0;
PowerPlant.LC.k.initial := -3.88889;
PowerPlant.LC.k.initial.initial := 0.0;
PowerPlant.CC.UnitConIn.Pin.initial := 691591.0;
PowerPlant.CC.UnitConIn.Win.initial := 40.7716;
PowerPlant.CC.UnitConIn.Hin.initial := 709.233;
PowerPlant.CC.UnitConOut.Pout.initial := 691425.0;
PowerPlant.CC.UnitConOut.Wout.initial := 41.9716;
PowerPlant.CC.FeulFlow.initial := 1.2;
PowerPlant.CC.FeulEnergy.initial := 43000.0;
PowerPlant.IC.HeatExWarmSideIn.Pin.initial := 264000.0;
PowerPlant.IC.HeatExWarmSideIn.Win.initial := 30.0;
PowerPlant.IC.HeatExWarmSideIn.Hin.initial := 109.102;
PowerPlant.IC.HeatExWarmSideOut.Pout.initial := 263100.0;
PowerPlant.IC.HeatExWarmSideOut.Wout.initial := 30.0;
PowerPlant.IC.HeatExWarmSideOut.Hout.initial := 100.0;
PowerPlant.IC.HeatExColdSideIn.Pin.initial := 1.2e+06;
PowerPlant.IC.HeatExColdSideIn.Win.initial := 100.0;
PowerPlant.IC.HeatExColdSideIn.Hin.initial := 73.711;
PowerPlant.IC.HeatExColdSideOut.Pout.initial := 1.19e+06;
PowerPlant.IC.HeatExColdSideOut.Wout.initial := 100.0;
PowerPlant.IC.Pinch1.initial := 273.05;
PowerPlant.IC.Pinch1.initial.initial := 0.0;
PowerPlant.IC.Pinch2.initial := 273.05;
PowerPlant.IC.Pinch2.initial.initial := 0.0;
PowerPlant.IC.E1.initial := 7371.1;
PowerPlant.IC.E1.initial.initial := 0.0;
PowerPlant.IC.E2.initial := 7644.15;
PowerPlant.IC.E2.initial.initial := 0.0;
PowerPlant.IC.E3.initial := 3273.05;
PowerPlant.IC.E3.initial.initial := 0.0;
PowerPlant.IC.E4.initial := 3000.0;
```

```
PowerPlant.IC.E4.initial.initial := 0.0;
PowerPlant.Gen.PowerIn.initial := 27328.8;
PowerPlant.Gen.P.initial := 27328.8;
PowerPlant.Gen.P.initial.initial := 0.0;
PowerPlant.Rec.HeatExWarmSideIn.Pin.initial := 100000.0;
PowerPlant.Rec.HeatExWarmSideIn.Win.initial := 41.9716;
PowerPlant.Rec.HeatExWarmSideIn.Hin.initial := 1167.1;
PowerPlant.Rec.HeatExWarmSideOut.Pout.initial := 98238.4;
PowerPlant.Rec.HeatExWarmSideOut.Wout.initial := 41.9716;
PowerPlant.Rec.HeatExColdSideIn.Pin.initial := 693254.0;
PowerPlant.Rec.HeatExColdSideIn.Win.initial := 40.7716;
PowerPlant.Rec.HeatExColdSideIn.Hin.initial := 113.52;
PowerPlant.Rec.HeatExColdSideOut.Pout.initial := 691591.0;
PowerPlant.Rec.HeatExColdSideOut.Wout.initial := 40.7716;
PowerPlant.Rec.Diff1.initial := -24288.2;
PowerPlant.Rec.Diff1.initial.initial := 0.0;
PowerPlant.Rec.Diff2.initial := 27163.7;
PowerPlant.Rec.Diff2.initial.initial := 0.0;
PowerPlant.Rec.Theta1.initial := 457.865;
PowerPlant.Rec.Theta1.initial.initial := 5000.0;
PowerPlant.Rec.Theta2.initial := 406.385;
PowerPlant.Rec.Theta2.initial.initial := 4000.0;
PowerPlant.Rec.Q.initial := 15414.8;
PowerPlant.Rec.Q.initial.initial := 0.0;
PowerPlant.Rec.E1.initial := 4628.37;
PowerPlant.Rec.E1.initial.initial := 0.0;
PowerPlant.Rec.E2.initial := 28916.6;
PowerPlant.Rec.E2.initial.initial := 0.0;
PowerPlant.Rec.E3.initial := 48985.0;
PowerPlant.Rec.E3.initial.initial := 0.0;
PowerPlant.Rec.E4.initial := 21821.2;
PowerPlant.Rec.E4.initial.initial := 0.0;
PowerPlant.Tower.Tray1.Air.AirSysIn.Pin.initial := 700000.0;
PowerPlant.Tower.Tray1.Air.AirSysIn.Win.initial := 30.0;
PowerPlant.Tower.Tray1.Air.AirSysIn.Hin.initial := 128.612;
PowerPlant.Tower.Tray1.Air.AirSysOut.Pout.initial := 698956.0;
PowerPlant.Tower.Tray1.Air.AirSysOut.Wout.initial := 32.3109;
PowerPlant.Tower.Tray1.Air.Houtg.initial := 124.729;
PowerPlant.Tower.Tray1.Air.Qout.initial := 7625.1;
PowerPlant.Tower.Tray1.Air.Qin.initial := 7800.01;
PowerPlant.Tower.Tray1.Air.mg.initial := 2.31093;
PowerPlant.Tower.Tray1.Air.Tin.initial := 71.4508;
PowerPlant.Tower.Tray1.Air.Tin.initial.initial := 0.0;
PowerPlant.Tower.Tray1.Air.Q.initial := 174.905;
PowerPlant.Tower.Tray1.Air.Q.initial.initial := 0.0;
PowerPlant.Tower.Tray1.Film.Qing.initial := 7800.01;
PowerPlant.Tower.Tray1.Film.Qoutg.initial := 7625.1;
PowerPlant.Tower.Tray1.Film.Qoutl.initial := 174.905;
PowerPlant.Tower.Tray1.Film.Houtg.initial := 124.729;
PowerPlant.Tower.Tray1.Film.Houtl.initial := 75.6861;
```

```
PowerPlant.Tower.Tray1.Film.mg.initial := 2.31093;
PowerPlant.Tower.Tray1.Film.ml.initial := 2.31093;
PowerPlant.Tower.Tray1.Wat.WatSysIn.Pin.initial := 1.1864e+06;
PowerPlant.Tower.Tray1.Wat.WatSysIn.Win.initial := 91.5394;
PowerPlant.Tower.Tray1.Wat.WatSysIn.Hin.initial := 75.8528;
PowerPlant.Tower.Tray1.Wat.WatSysOut.Pout.initial := 1.18561e+06;
PowerPlant.Tower.Tray1.Wat.WatSysOut.Wout.initial := 89.2284;
PowerPlant.Tower.Tray1.Wat.Houtl.initial := 75.6861;
PowerPlant.Tower.Tray1.Wat.ml.initial := 2.31093;
PowerPlant.Tower.Tray1.Wat.Qout.initial := 174.905;
PowerPlant.Tower.Tray1.AirIn.Pin.initial := 700000.0;
PowerPlant.Tower.Tray1.AirIn.Win.initial := 30.0;
PowerPlant.Tower.Tray1.AirIn.Hin.initial := 128.612;
PowerPlant.Tower.Tray1.WatIn.Pin.initial := 1.1864e+06;
PowerPlant.Tower.Tray1.WatIn.Win.initial := 91.5394;
PowerPlant.Tower.Tray1.WatIn.Hin.initial := 75.8528;
PowerPlant.Tower.Tray1.AirOut.Pout.initial := 698956.0;
PowerPlant.Tower.Tray1.AirOut.Wout.initial := 32.3109;
PowerPlant.Tower.Tray1.AirOut.Hout.initial := 124.729;
PowerPlant.Tower.Tray1.WatOut.Pout.initial := 1.18561e+06;
PowerPlant.Tower.Tray1.WatOut.Wout.initial := 89.2284;
PowerPlant.Tower.Tray1.WatOut.Hout.initial := 75.6861;
PowerPlant.Tower.Tray2.Air.AirSysIn.Pin.initial := 698956.0;
PowerPlant.Tower.Tray2.Air.AirSysIn.Win.initial := 32.3109;
PowerPlant.Tower.Tray2.Air.AirSysIn.Hin.initial := 124.729;
PowerPlant.Tower.Tray2.Air.AirSysOut.Pout.initial := 697763.0;
PowerPlant.Tower.Tray2.Air.AirSysOut.Wout.initial := 34.5339;
PowerPlant.Tower.Tray2.Air.Houtg.initial := 121.404;
PowerPlant.Tower.Tray2.Air.Qout.initial := 7341.89;
PowerPlant.Tower.Tray2.Air.Qin.initial := 7510.51;
PowerPlant.Tower.Tray2.Air.mg.initial := 2.22297;
PowerPlant.Tower.Tray2.Air.Tin.initial := 69.294;
PowerPlant.Tower.Tray2.Air.Tin.initial.initial := 0.0;
PowerPlant.Tower.Tray2.Air.Q.initial := 168.618;
PowerPlant.Tower.Tray2.Air.Q.initial.initial := 0.0;
PowerPlant.Tower.Tray2.Film.Qing.initial := 7510.51;
PowerPlant.Tower.Tray2.Film.Qoutg.initial := 7341.89;
PowerPlant.Tower.Tray2.Film.Qoutl.initial := 168.618;
PowerPlant.Tower.Tray2.Film.Houtg.initial := 121.404;
PowerPlant.Tower.Tray2.Film.Houtl.initial := 75.8528;
PowerPlant.Tower.Tray2.Film.mg.initial := 2.22297;
PowerPlant.Tower.Tray2.Film.ml.initial := 2.22297;
PowerPlant.Tower.Tray2.Wat.WatSysIn.Pin.initial := 1.18724e+06;
PowerPlant.Tower.Tray2.Wat.WatSysIn.Win.initial := 93.7623;
PowerPlant.Tower.Tray2.Wat.WatSysIn.Hin.initial := 76.0109;
PowerPlant.Tower.Tray2.Wat.WatSysOut.Pout.initial := 1.1864e+06;
PowerPlant.Tower.Tray2.Wat.WatSysOut.Wout.initial := 91.5394;
PowerPlant.Tower.Tray2.Wat.Houtl.initial := 75.8528;
PowerPlant.Tower.Tray2.Wat.ml.initial := 2.22297;
PowerPlant.Tower.Tray2.Wat.Qout.initial := 168.618;
```

```
PowerPlant.Tower.Tray2.AirIn.Pin.initial := 698956.0;
PowerPlant.Tower.Tray2.AirIn.Win.initial := 32.3109;
PowerPlant.Tower.Tray2.AirIn.Hin.initial := 124.729;
PowerPlant.Tower.Tray2.WatIn.Pin.initial := 1.18724e+06;
PowerPlant.Tower.Tray2.WatIn.Win.initial := 93.7623;
PowerPlant.Tower.Tray2.WatIn.Hin.initial := 76.0109;
PowerPlant.Tower.Tray2.AirOut.Pout.initial := 697763.0;
PowerPlant.Tower.Tray2.AirOut.Wout.initial := 34.5339;
PowerPlant.Tower.Tray2.AirOut.Hout.initial := 121.404;
PowerPlant.Tower.Tray2.WatOut.Pout.initial := 1.1864e+06;
PowerPlant.Tower.Tray2.WatOut.Wout.initial := 91.5394;
PowerPlant.Tower.Tray2.WatOut.Hout.initial := 75.8528;
PowerPlant.Tower.Tray3.Air.AirSysIn.Pin.initial := 697763.0;
PowerPlant.Tower.Tray3.Air.AirSysIn.Win.initial := 34.5339;
PowerPlant.Tower.Tray3.Air.AirSysIn.Hin.initial := 121.404;
PowerPlant.Tower.Tray3.Air.AirSysOut.Pout.initial := 696418.0;
PowerPlant.Tower.Tray3.Air.AirSysOut.Wout.initial := 36.6799;
PowerPlant.Tower.Tray3.Air.Houtg.initial := 118.493;
PowerPlant.Tower.Tray3.Air.Qout.initial := 7093.51;
PowerPlant.Tower.Tray3.Air.Qin.initial := 7256.63;
PowerPlant.Tower.Tray3.Air.mg.initial := 2.14598;
PowerPlant.Tower.Tray3.Air.Tin.initial := 67.4466;
PowerPlant.Tower.Tray3.Air.Tin.initial.initial := 0.0;
PowerPlant.Tower.Tray3.Air.Q.initial := 163.118;
PowerPlant.Tower.Tray3.Air.Q.initial.initial := 0.0;
PowerPlant.Tower.Tray3.Film.Qing.initial := 7256.63;
PowerPlant.Tower.Tray3.Film.Qoutg.initial := 7093.51;
PowerPlant.Tower.Tray3.Film.Qoutl.initial := 163.118;
PowerPlant.Tower.Tray3.Film.Houtg.initial := 118.493;
PowerPlant.Tower.Tray3.Film.Houtl.initial := 76.0109;
PowerPlant.Tower.Tray3.Film.mg.initial := 2.14598;
PowerPlant.Tower.Tray3.Film.ml.initial := 2.14598;
PowerPlant.Tower.Tray3.Wat.WatSysIn.Pin.initial := 1.18812e+06;
PowerPlant.Tower.Tray3.Wat.WatSysIn.Win.initial := 95.9083;
PowerPlant.Tower.Tray3.Wat.WatSysIn.Hin.initial := 76.1613;
PowerPlant.Tower.Tray3.Wat.WatSysOut.Pout.initial := 1.18724e+06;
PowerPlant.Tower.Tray3.Wat.WatSysOut.Wout.initial := 93.7623;
PowerPlant.Tower.Tray3.Wat.Houtl.initial := 76.0109;
PowerPlant.Tower.Tray3.Wat.ml.initial := 2.14598;
PowerPlant.Tower.Tray3.Wat.Qout.initial := 163.118;
PowerPlant.Tower.Tray3.AirIn.Pin.initial := 697763.0;
PowerPlant.Tower.Tray3.AirIn.Win.initial := 34.5339;
PowerPlant.Tower.Tray3.AirIn.Hin.initial := 121.404;
PowerPlant.Tower.Tray3.WatIn.Pin.initial := 1.18812e+06;
PowerPlant.Tower.Tray3.WatIn.Win.initial := 95.9083;
PowerPlant.Tower.Tray3.WatIn.Hin.initial := 76.1613;
PowerPlant.Tower.Tray3.AirOut.Pout.initial := 696418.0;
PowerPlant.Tower.Tray3.AirOut.Wout.initial := 36.6799;
PowerPlant.Tower.Tray3.AirOut.Hout.initial := 118.493;
PowerPlant.Tower.Tray3.WatOut.Pout.initial := 1.18724e+06;
```

```
PowerPlant.Tower.Tray3.WatOut.Wout.initial := 93.7623;
PowerPlant.Tower.Tray3.WatOut.Hout.initial := 76.0109;
PowerPlant.Tower.Tray4.Air.AirSysIn.Pin.initial := 696418.0;
PowerPlant.Tower.Tray4.Air.AirSysIn.Win.initial := 36.6799;
PowerPlant.Tower.Tray4.Air.AirSysIn.Hin.initial := 118.493;
PowerPlant.Tower.Tray4.Air.AirSysOut.Pout.initial := 694916.0;
PowerPlant.Tower.Tray4.Air.AirSysOut.Wout.initial := 38.7571;
PowerPlant.Tower.Tray4.Air.Houtg.initial := 115.891;
PowerPlant.Tower.Tray4.Air.Qout.initial := 6871.23;
PowerPlant.Tower.Tray4.Air.Qin.initial := 7029.43;
PowerPlant.Tower.Tray4.Air.mg.initial := 2.07719;
PowerPlant.Tower.Tray4.Air.Tin.initial := 65.8293;
PowerPlant.Tower.Tray4.Air.Tin.initial.initial := 0.0;
PowerPlant.Tower.Tray4.Air.Q.initial := 158.201;
PowerPlant.Tower.Tray4.Air.Q.initial.initial := 0.0;
PowerPlant.Tower.Tray4.Film.Qing.initial := 7029.43;
PowerPlant.Tower.Tray4.Film.Qoutg.initial := 6871.23;
PowerPlant.Tower.Tray4.Film.Qoutl.initial := 158.201;
PowerPlant.Tower.Tray4.Film.Houtg.initial := 115.891;
PowerPlant.Tower.Tray4.Film.Houtl.initial := 76.1613;
PowerPlant.Tower.Tray4.Film.mg.initial := 2.07719;
PowerPlant.Tower.Tray4.Film.ml.initial := 2.07719;
PowerPlant.Tower.Tray4.Wat.WatSysIn.Pin.initial := 1.18904e+06;
PowerPlant.Tower.Tray4.Wat.WatSysIn.Win.initial := 97.9855;
PowerPlant.Tower.Tray4.Wat.WatSysIn.Hin.initial := 76.3046;
PowerPlant.Tower.Tray4.Wat.WatSysOut.Pout.initial := 1.18812e+06;
PowerPlant.Tower.Tray4.Wat.WatSysOut.Wout.initial := 95.9083;
PowerPlant.Tower.Tray4.Wat.Houtl.initial := 76.1613;
PowerPlant.Tower.Tray4.Wat.ml.initial := 2.07719;
PowerPlant.Tower.Tray4.Wat.Qout.initial := 158.201;
PowerPlant.Tower.Tray4.AirIn.Pin.initial := 696418.0;
PowerPlant.Tower.Tray4.AirIn.Win.initial := 36.6799;
PowerPlant.Tower.Tray4.AirIn.Hin.initial := 118.493;
PowerPlant.Tower.Tray4.WatIn.Pin.initial := 1.18904e+06;
PowerPlant.Tower.Tray4.WatIn.Win.initial := 97.9855;
PowerPlant.Tower.Tray4.WatIn.Hin.initial := 76.3046;
PowerPlant.Tower.Tray4.AirOut.Pout.initial := 694916.0;
PowerPlant.Tower.Tray4.AirOut.Wout.initial := 38.7571;
PowerPlant.Tower.Tray4.AirOut.Hout.initial := 115.891;
PowerPlant.Tower.Tray4.WatOut.Pout.initial := 1.18812e+06;
PowerPlant.Tower.Tray4.WatOut.Wout.initial := 95.9083;
PowerPlant.Tower.Tray4.WatOut.Hout.initial := 76.1613;
PowerPlant.Tower.Tray5.Air.AirSysIn.Pin.initial := 694916.0;
PowerPlant.Tower.Tray5.Air.AirSysIn.Win.initial := 38.7571;
PowerPlant.Tower.Tray5.Air.AirSysIn.Hin.initial := 115.891;
PowerPlant.Tower.Tray5.Air.AirSysOut.Pout.initial := 693254.0;
PowerPlant.Tower.Tray5.Air.AirSysOut.Wout.initial := 40.7716;
PowerPlant.Tower.Tray5.Air.Houtg.initial := 113.52;
PowerPlant.Tower.Tray5.Air.Qout.initial := 6668.34;
PowerPlant.Tower.Tray5.Air.Qin.initial := 6822.06;
```

```
PowerPlant.Tower.Tray5.Air.mg.initial := 2.0145;
PowerPlant.Tower.Tray5.Air.Tin.initial := 64.3841;
PowerPlant.Tower.Tray5.Air.Tin.initial.initial := 0.0;
PowerPlant.Tower.Tray5.Air.Q.initial := 153.715;
PowerPlant.Tower.Tray5.Air.Q.initial.initial := 0.0;
PowerPlant.Tower.Tray5.Film.Qing.initial := 6822.06;
PowerPlant.Tower.Tray5.Film.Qoutg.initial := 6668.34;
PowerPlant.Tower.Tray5.Film.Qoutl.initial := 153.715;
PowerPlant.Tower.Tray5.Film.Houtg.initial := 113.52;
PowerPlant.Tower.Tray5.Film.Houtl.initial := 76.3046;
PowerPlant.Tower.Tray5.Film.mg.initial := 2.0145;
PowerPlant.Tower.Tray5.Film.ml.initial := 2.0145;
PowerPlant.Tower.Tray5.Wat.WatSysIn.Pin.initial := 1.19e+06;
PowerPlant.Tower.Tray5.Wat.WatSysIn.Win.initial := 100.0;
PowerPlant.Tower.Tray5.Wat.WatSysIn.Hin.initial := 76.4415;
PowerPlant.Tower.Tray5.Wat.WatSysOut.Pout.initial := 1.18904e+06;
PowerPlant.Tower.Tray5.Wat.WatSysOut.Wout.initial := 97.9855;
PowerPlant.Tower.Tray5.Wat.Houtl.initial := 76.3046;
PowerPlant.Tower.Tray5.Wat.ml.initial := 2.0145;
PowerPlant.Tower.Tray5.Wat.Qout.initial := 153.715;
PowerPlant.Tower.Tray5.AirIn.Pin.initial := 694916.0;
PowerPlant.Tower.Tray5.AirIn.Win.initial := 38.7571;
PowerPlant.Tower.Tray5.AirIn.Hin.initial := 115.891;
PowerPlant.Tower.Tray5.WatIn.Pin.initial := 1.19e+06;
PowerPlant.Tower.Tray5.WatIn.Win.initial := 100.0;
PowerPlant.Tower.Tray5.WatIn.Hin.initial := 76.4415;
PowerPlant.Tower.Tray5.AirOut.Pout.initial := 693254.0;
PowerPlant.Tower.Tray5.AirOut.Wout.initial := 40.7716;
PowerPlant.Tower.Tray5.AirOut.Hout.initial := 113.52;
PowerPlant.Tower.Tray5.WatOut.Pout.initial := 1.18904e+06;
PowerPlant.Tower.Tray5.WatOut.Wout.initial := 97.9855;
PowerPlant.Tower.Tray5.WatOut.Hout.initial := 76.3046;
PowerPlant.Tower.TowerAirIn.Pin.initial := 700000.0;
PowerPlant.Tower.TowerAirIn.Win.initial := 30.0;
PowerPlant.Tower.TowerAirIn.Hin.initial := 128.612;
PowerPlant.Tower.TowerAirOut.Pout.initial := 693254.0;
PowerPlant.Tower.TowerAirOut.Wout.initial := 40.7716;
PowerPlant.Tower.TowerAirOut.Hout.initial := 113.52;
PowerPlant.Tower.TowerWatIn.Pin.initial := 1.19e+06;
PowerPlant.Tower.TowerWatIn.Win.initial := 100.0;
PowerPlant.Tower.TowerWatIn.Hin.initial := 76.4415;
PowerPlant.Tower.TowerWatOut.Pout.initial := 1.18561e+06;
PowerPlant.Tower.TowerWatOut.Wout.initial := 89.2284;
PowerPlant.Tower.TowerWatOut.Hout.initial := 75.6861;
PowerPlant.Tower.dTin.initial := 37.0781;
PowerPlant.Tower.dTin.initial.initial := 0.0;
PowerPlant.Tower.dTout.initial := 52.9254;
PowerPlant.Tower.dTout.initial.initial := 0.0;
PowerPlant.Eq.e.initial := 42.7294;
PowerPlant.Eq.Wr.initial := 427.294;
```

```
PowerPlant.Eq.InFlow.Pin.initial := 1.18561e+06;
PowerPlant.Eq.InFlow.Win.initial := 89.2284;
PowerPlant.Eq.InFlow.Hin.initial := 75.6861;
PowerPlant.Eq.OutFlow.Pout.initial := 1.2e+06;
PowerPlant.Eq.OutFlow.Wout.initial := 100.0;
PowerPlant.Eq.m.initial.initial := 1000.0;
PowerPlant.Eq.V.initial := 7.47884;
PowerPlant.Eq.V.initial.initial := 50.0;
PowerPlant.Eq.P.initial := 1.18561e+06;
PowerPlant.Eq.P.initial.initial := 0.0;
PowerPlant.Split.InFlow.Pin.initial := 693254.0;
PowerPlant.Split.InFlow.Win.initial := 40.7716;
PowerPlant.Split.InFlow.Hin.initial := 113.52;
PowerPlant.Split.OutFlow.Pout.initial := 693254.0;
PowerPlant.Split.OutFlow.Wout.initial := 40.7716;
PowerPlant.Split.OutFlow.Hout.initial := 113.52;
PowerPlant.Split.CoolAirFlow.initial := 0.0;
PowerPlant.Split.CoolAirEnth.initial := 113.52;
PowerPlant.Split.X.initial := 0.0;
PowerPlant.TempCon.U.initial := 0.0;
PowerPlant.TempCon.H1.initial := 1918.36;
PowerPlant.TempCon.H2.initial := 1918.85;
PowerPlant.TempCon.X.initial := 0.0;
PowerPlant.Lambda.initial := 0.564049;
PowerPlant.Lambda.initial.initial := 0.0;
PowerPlant.Eta.initial := 0.529627;
PowerPlant.Eta.initial.initial := 0.0;
PowerPlant.ART.initial := 0.792771;
PowerPlant.ART.initial.initial := 0.0;
PowerPlant.EUF.initial := 0.828363;
PowerPlant.EUF.initial.initial := 0.0;
PowerPlant.FESR.initial := 0.396134;
PowerPlant.FESR.initial.initial := 0.0;
PowerPlant.IHR.initial := 1.2614;
PowerPlant.IHR.initial.initial := 0.0;
```