

Diss. ETH No. 14594

# Fault Diagnosis for Heat Pump Systems

A dissertation submitted to the  
SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH

for the degree of  
Doctor of Technical Sciences

presented by  
David Zogg  
Dipl. Masch.-Ing. ETH  
born January 15, 1973  
citizen of Grabs SG

accepted on the recommendation of  
Prof. Dr. H. P. Geering, examiner  
Prof. Dr. A. H. Glattfelder, co-examiner  
Dr. E. Shafai, co-examiner

Zurich 2002

ISBN 3-906483-02-9

IMRT Press

c/o Institut für Mess- und Regeltechnik

ETH Zentrum

8092 Zürich

Schweiz

06/2002

To my parents



## **Preface**

This thesis is based on research performed at the Measurement and Control Laboratory (IMRT) of the Swiss Federal Institute of Technology (ETH) in Zurich between 1998 and 2002. It is part of the project ‘Short-Term Rating Method’ of the Swiss Federal Office of Energy (BFE), project No. 33684. The Swiss Federal Office of Energy is gratefully acknowledged for providing a significant part of the necessary funding.

First of all I would like to thank my supervisor, Professor Dr. H. P. Geering, for having accepted and supported my research project in the field of fault detection and diagnosis, as well as Prof. Dr. A. H. Glattfelder for his assistance as co-examiner. Especially I would like to thank Dr. E. Shafai, the initiator of the Heat Pump Group, for managing the project and for his advice during the research.

Sincere thanks are extended to Hans Ulrich Honegger and Oskar Brachs for their skilfull support in building the heat pump test bench at our Laboratory, as well as to Roger Wimmer and Simon Ginsburg for the valuable discussions within the Heat Pump Group.

I am also grateful to Dr. H. U. Bruderer and H. Hohl of SATAG Thermo-technik AG for their technical support and for providing a specially equipped commercial heat pump for the test bench, to Dr. J. Tödli of Siemens Building Technologies AG, to H. R. Gabathuler and H. Mayer of Gabathuler AG, and to all other partners of the project for their helpful assistance.

Special thanks are also addressed to Prof. M. Conzett and E. Blattmann of the University of Applied Sciences, Winterthur (ZHW) for providing their Laboratory heat pump, which was an important first step for testing my methods.

Finally, I wish to thank Brigitte Rohrbach for her editing of the text and all other people of the Institute who supported me during the doctoral studies. My greatest appreciation is reserved for my family.



## Abstract

For mitigating global warming, the energy consumption and CO<sub>2</sub> emission of heating, refrigerating, and air-conditioning systems has to be minimized. Therefore, the COP (coefficient of performance) of heat pumps has to be at a maximum during operation. With the fault diagnosis methods developed, a degradation of the performance is detected early, and the exact reasons for the degradation are diagnosed. Two diagnosis systems are developed for operational monitoring of heat pump systems. They allow a condition-oriented maintenance, which also reduces service costs. The fault diagnosis systems are tested with data from different heat pump applications, such as measured data from a residential building, data from a heat pump simulation model, and data from two test benches.

Since the additional hardware costs should be minimal, the first fault diagnosis system is designed to require only a small number of sensors. During operation, the parameters of a dynamic gray-box model are sequentially re-identified. The faults are then classified from these parameters, using statistical techniques as well as techniques from fuzzy logic and neural network theory. Before operation, the diagnosis system is trained on the basis of experimental data for known fault cases. During the training phase, clusters are built in the parameter space for each fault case. With a small number of sensors, a large number of fault cases is classified with a high classification quality.

The second diagnosis system is designed for a fast commercial realization because of its simple structure and the very small training effort. During operation, the parameters of a steady-state physical model are sequentially re-identified. The faults then are interpreted by the service technician or by the user, which is straightforward because of the physical meaning of the parameters. Before operation, the diagnosis system is initialized by nominal (fault-free) data.

For testing the diagnosis systems, a detailed physical heat pump simulation model is developed, which covers all operating modes. Various fault cases are simulated by changing the corresponding model parameters. In addition, a new test bench was built, which contains a modified commercial heat pump. With the test bench, data is generated for several fault cases.

With the fault diagnosis systems developed, the energetic efficiencies of heat pumps as well as larger heating, refrigerating, or air-conditioning systems can be significantly improved.



## **Kurzfassung**

Um der globalen Erwärmung entgegenzuwirken, muss der Energieverbrauch und damit die CO<sub>2</sub>-Emission von Heiz-, Kühl- und Klimatisierungsanlagen minimiert werden. Deshalb sollte auch die Leistungsziffer (COP, coefficient of performance) von Wärmepumpen während dem Betrieb maximal sein. Mit den hier entwickelten Fehlerdiagnosemethoden kann eine Abnahme der Leistungsziffer frühzeitig detektiert werden, und die genauen Gründe der Abnahme können diagnostiziert werden. Für die Betriebsüberwachung von Wärmepumpensystemen wurden zwei Diagnosemethoden entwickelt, welche eine zustandsorientierte Instandhaltung mit tieferen Servicekosten ermöglichen. Sie wurden mit Messdaten eines Einfamilienhauses, simulierten Daten eines Wärmepumpen-Modelles und Messdaten von zwei Prüfständen getestet.

Um die zusätzlichen Hardware-Kosten minimal zu halten, benötigt das erste Überwachungssystem nur wenige Sensoren. Während dem Betrieb werden die Parameter eines dynamischen „Gray-Box“-Modelles laufend identifiziert und daraus die Fehler klassifiziert. Für die Fehlerklassifikation werden sowohl statistische Methoden als auch Methoden der „Fuzzy Logic“ und der neuronalen Netze verwendet. Vor dem Betrieb wird das Diagnosesystem mit experimentellen Daten für bekannte Fehlerfälle trainiert. Während der Trainingsphase werden im Parameterraum „Cluster“ gebildet für jeden Fehlerfall. Bereits mit einer geringen Anzahl Sensoren kann eine grosse Anzahl von Fehlerfällen mit einer hohen Klassifikationsgüte erkannt werden.

Das zweite Diagnosesystem ist wegen seiner einfachen Struktur und dem sehr geringen Trainingsaufwand für eine schnelle industrielle Umsetzung bestimmt. Während dem Betrieb werden die Parameter eines stationären physikalischen Modelles laufend identifiziert. Daraus können die Fehler durch den Servicetechniker oder Benutzer interpretiert werden. Aufgrund der physikalischen Bedeutung der Parameter ist dies auf einfache Art und Weise möglich. Vor dem Betrieb wird das Diagnosesystem mit Nominaldaten initialisiert (für den fehlerfreien Fall).

Um die Diagnosesysteme zu testen, wurde ein detailliertes physikalisches Wärmepumpen-Simulationsmodell entwickelt, welches alle Betriebsmodi und zahlreiche Fehlerfälle abdeckt. Zusätzlich wurde ein neuer Prüfstand entwickelt, welcher eine modifizierte industrielle Wärmepumpe enthält.

Mit den entwickelten Diagnosesystemen können die energetischen Wirkungsgrade von Wärmepumpen und von grösseren Heizungs-, Kühlungs- oder Klimatisierungsanlagen markant verbessert werden.



# Contents

<b>1</b>	<b>Symbols .....</b>	<b>17</b>
<b>2</b>	<b>Introduction .....</b>	<b>21</b>
2.1	Motivation.....	21
2.2	The Heat Pump Market.....	23
2.3	The Project ‘Short-Term Rating Method’ .....	24
2.4	Separation to other Projects.....	25
2.5	Chronology of Development.....	26
2.6	Overview.....	28
<b>3</b>	<b>An Introduction to Fault Detection and Diagnosis.....</b>	<b>33</b>
3.1	Definitions.....	33
3.2	Methods .....	33
3.3	State-of-the-Art Applications to HVAC Systems .....	35
<b>4</b>	<b>The Heat Pump .....</b>	<b>37</b>
4.1	The Refrigerant Cycle.....	37
4.2	Potential Faults.....	40
<b>5</b>	<b>A Fault Diagnosis System with Integrated Fault Classification – Theory .....</b>	<b>43</b>
5.1	Concept.....	43
5.1.1	Training Procedure.....	44
5.1.2	Operation.....	46
5.2	Model.....	47
5.2.1	Physical Structure (Gray-Box Model) .....	48
5.2.2	Decoupled Structure (Black-Box Model) .....	48
5.2.3	Full Structure (Black-Box Model).....	49
5.2.4	Submodels .....	50
5.3	Parameter Identification.....	52
5.3.1	Least-Squares Estimation .....	53
5.3.2	Signal Plots for Submodels.....	54
5.3.3	Model Error Analysis .....	58
5.3.4	Sensitivity Analysis.....	60

5.4	Fault Classification Training .....	63
5.4.1	HCM2Fuzzy Method.....	64
5.4.2	HCM2Neuro Method.....	66
5.4.3	FCM2Neuro and PFCM2Neuro Methods .....	68
5.4.4	Cluster Checking.....	72
5.5	Minimizing the Number of Sensors .....	73
5.5.1	Checking Combinations of Submodels .....	73
5.5.2	Reduction of Model Structure .....	75
5.6	Gradual and Simultaneous Faults .....	76
5.6.1	Standard Clusters .....	77
5.6.2	Vector Clusters.....	78
5.7	Online Fault Diagnosis.....	80
5.7.1	Diagnosis Steps.....	80
5.7.2	Averaging .....	81
5.7.3	Fault Trend Charts.....	82
5.7.4	False Alarm Tuning.....	82
<b>6</b>	<b>A Fault Diagnosis System with Integrated Fault Classification – Applications .....</b>	<b>85</b>
6.1	Data from ZHW Test Bench (Application II).....	85
6.1.1	Configuration of the Test Heat Pump .....	86
6.1.2	Model .....	88
6.1.3	Signal Plots for the Physical Model Structure .....	88
6.1.4	Error Charts for the Physical Model Structure.....	93
6.1.5	Parameter Charts for the Physical Model Structure .....	94
6.1.6	Signal Plots for the Decoupled Model Structure.....	95
6.1.7	Error Charts for the Decoupled Model Structure .....	97
6.1.8	Standard Clusters .....	98
6.1.9	Classification Results for Standard Clusters.....	100
6.2	Data from Simulation Model (Application III).....	103
6.2.1	Configuration of the Simulated Heat Pump.....	103
6.2.2	Model .....	106
6.2.3	Signal Plots .....	106
6.2.4	Error Charts .....	108
6.2.5	Classification Results for Standard Clusters.....	110
6.2.6	Classification Results for Vector Clusters.....	111

6.3	Data from IMRT Test Bench (Application IV) .....	115
6.3.1	Configuration of the Test Heat Pump .....	115
6.3.2	Model .....	117
6.3.3	Signal Plots for Two Output Signals .....	118
6.3.4	Error Charts and Parameter Charts for Two Output Signals .....	120
6.3.5	Signal Plots for Three Output Signals .....	122
6.3.6	Classification Results for Standard Clusters .....	123
6.3.7	Classification Results for Vector Clusters .....	124
6.3.8	False Alarm Tuning.....	124
6.3.9	Simultaneous Faults .....	125
<b>7</b>	<b>Software FuzzyWatch.....</b>	<b>127</b>
7.1	Training Procedure .....	127
7.2	Online Diagnosis .....	131
7.2.1	Properties for Online Diagnosis .....	132
7.2.2	Sequential Classification with Standard Clusters .....	132
7.2.3	Sequential Classification with Vector Clusters .....	134
7.2.4	Trend Charts .....	138
<b>8</b>	<b>A Fault Diagnosis System with Steady-State Parameter Identification – Theory .....</b>	<b>139</b>
8.1	Concept.....	139
8.1.1	Initialization.....	140
8.1.2	Operation.....	140
8.2	Modeling and Parameter Identification.....	141
8.2.1	Sensors Required .....	141
8.2.2	Steady-State Equations.....	143
8.2.3	Linear Regression Form .....	144
8.2.4	Parameter Identification .....	145
8.3	Steady-State Detection.....	145
8.4	Additional Parameters and Characteristics.....	147
8.5	Interpretation of the Parameters .....	149
8.6	Gradual and Simultaneous Faults .....	150
8.7	Online Fault Diagnosis .....	150
8.7.1	Diagnosis Steps.....	150
8.7.2	Averaging.....	151
8.7.3	Parameter Trend Charts.....	151

<b>9</b>	<b>A Fault Diagnosis System with Steady-State Parameter Identification – Applications</b> .....	<b>153</b>
9.1	Data from Residential Building (Application I) .....	153
9.2	Data from IMRT Test Bench (Application IV) .....	159
9.2.1	Configuration of the Test Heat Pump .....	159
9.2.2	Diagnosis Results .....	160
<b>10</b>	<b>Software HeatWatch</b> .....	<b>163</b>
10.1	Initialization and Properties.....	165
10.2	Online Diagnosis .....	166
10.2.1	Sequential Parameter Identification.....	167
10.2.2	Trend Charts .....	170
<b>11</b>	<b>Simulation Model</b> .....	<b>173</b>
11.1	Modeling the Heat Pump Refrigerant Cycle .....	173
11.1.1	Evaporation.....	175
11.1.2	Superheating .....	179
11.1.3	Compression .....	180
11.1.4	Hot Gas Cooling.....	181
11.1.5	Condensation .....	182
11.1.6	Subcooling .....	184
11.1.7	Expansion .....	185
11.1.8	Air Side .....	186
11.1.9	Water Side .....	187
11.1.10	Icing of the Evaporator .....	187
11.1.11	Refrigerant Data.....	190
11.1.12	Mean Temperatures and Simplifications.....	192
11.1.13	Implementation and Parameter Sets.....	193
11.2	Simulations.....	195
11.2.1	Validation for the Nominal Case.....	195
11.2.2	Coupling with Building .....	199
11.2.3	Simulations for Fault Cases .....	200
<b>12</b>	<b>IMRT Test Bench</b> .....	<b>205</b>
12.1	Test Environment .....	207
12.2	Data Acquisition and Automation .....	209
12.3	Running Test Cycles .....	210
12.3.1	Definition of the Test Cycles .....	210
12.3.2	Operating Conditions.....	211
12.3.3	Emulation .....	211

<b>13</b>	<b>Summary and Conclusions.....</b>	<b>213</b>
<b>14</b>	<b>Outlook .....</b>	<b>217</b>
14.1	Suggestions for the Realization Processes .....	217
14.1.1	FuzzyWatch.....	217
14.1.2	HeatWatch.....	219
14.2	Next Steps .....	221
<b>15</b>	<b>References.....</b>	<b>225</b>
<b>Appendix A: Identification .....</b>		
	Persistent Excitation and Consistency Properties .....	231
	Recursive Least-Squares Identification .....	232
<b>Appendix B: From Residuals to Parameters .....</b>		
<b>Appendix C: Data Sheets .....</b>		
	SATAG Natura AW110.....	239
	SATAG Natura BW108 .....	240
	ZHW Laboratory Heat Pump.....	241
	Simulation Model Parameters.....	243



# 1 Symbols

## *Symbols and units*

$A$	$m^2$	(heat transfer) area
$A_i, B_i, C_i, K, X$		parameters (refrigerant data)
$a_{i,k}, b_{i,k}$		model parameters (ARX approach)
$b$	$m$	(heat transfer) width
$c, c_p$	$J/kg \cdot K$	specific heat capacity
$c_{cmp}$	-	polytropic exponent of compressor
$c_{i,k}$		cluster center
$cm^*$	$W/K$	convective heat transfer parameter $c \cdot m^*$
$COP$	-	coefficient of performance
$d, d_i(x)$		distance
$e(t)$		white noise (linear regression)
$E$	$J$	energy
$E$		equation error vector (linear regression)
$f(i_{seq})$		fault set
$f_i, \Delta f_i$		fault size
$f(x), f_i(x)$		membership grade function
$g(x), g_i(x)$		fault grade function
$g_{min}$		minimal fault grade
$h$	$J/kg$	specific enthalpy
$i$		index
$k_{i,k}, K$		model parameters (static approach)
$k$	$W/m^2 \cdot K$	(heat transfer) coefficient
$kA$	$W/K$	heat transfer parameter $k \cdot A$
$kb$	$W/m \cdot K$	heat transfer parameter $k \cdot b$
$k_{cmp}$	$kg/sec \cdot Pa^{1/2}$	valve constant (backflow in the compressor)
$k_{exp}$	$kg/sec \cdot Pa^{1/2}$	valve constant of expansion valve
$k_{hg}$	$kg/sec \cdot Pa^{1/2}$	valve constant of hot-gas valve
$k_{hg,c}$	-	condensating refrigerant ratio (hot-gas valve)
$k_{ice}$		heat transfer reduction factor (evaporator)
$k_{ice,...}$		parameters of the icing process (evaporator)
$k_l$	-	liquid refrigerant ratio (expansion valve)
$k_R$	-	ratio of air and vapour gas constants
$k_{tr}$	$kg/sec \cdot Pa$	mass transfer rate (evaporator)
$l$		number of sequences
$L$	$m$	(evaporation) length

$m$		cluster parameter (FCM, PFCM clusters)
$m$	kg	mass
$m^*$	kg/s	mass flow
$n$	1/sec	rotational speed
$n_{...}$		number of ...
$p$	Pa, bar	pressure
$P$	W	power / power input
$PF$	-	energetic efficiency
$Q^*$	W	heat flow
$R$	J/kg	special gas constant
$r$	J/kg	specific latent heat of vaporization
$r_{ice}$	J/kg	specific latent heat of liquefaction
$r_i$		cluster radius (PFCM clusters)
$r_{wrong}$		rate of wrong classifications
$t$	sec	time
$T$	°C	temperature
$u$	J/kg	specific energy
$u, u(t), u_i, u_i(t)$		input signal / vector
$u_{step}$		step input signal
$v$	m <sup>3</sup> /kg	specific volume
$V$	m <sup>3</sup>	volume
$V^*$	m <sup>3</sup> /s	volumetric flow
$w$	m/sec	velocity
$w_{nom}$		nominal weighting factor
$x$		state vector
$x, x(\varphi)$		vapour mass ratio
$x_k, x(i_{seq})$		data point in parameter space (= parameter vector)
$y, y(t), y_i, y_i(t)$		output signal / vector
$Y$		output vector (linear regression)
$z$		vector of the static variables
$\eta$	-	efficiency
$\lambda$	-	mass flow supply efficiency
$\rho$	kg/m <sup>3</sup>	density
$\beta_e$	-	liquid area ratio (evaporation)
$\varepsilon$	-	exergetic efficiency
$\varepsilon(t)$		equation error (linear regression)
$\varphi$		regression vector (linear regression)
$\varphi$	-	relative humidity

$\Phi$		regression matrix (linear regression)
$\kappa, \kappa(i_{seq})$		(steady-state) parameter vector
$\theta, \theta_k, \theta(i_{seq})$		parameter / parameter vector
$\sigma, \sigma_{i,k}$		standard deviation
$\tau$	1/sec	time constant
$\Delta t$	sec	time interval
$\Delta T$	°C	temperature difference
$\Delta p$	Pa, bar	pressure difference
$\Delta \kappa$		parameter deviation (steady-state)
$\Delta \theta_0$		parameter bias
$\Delta \bar{\theta}$		parameter deviation
$\delta \theta$		parameter uncertainty

### **Indices**

...'	saturated liquid point (refrigerant data)
...''	saturated vapour point (refrigerant data)
0	initial value / operating point
0	off mode (shut-down)
1	on mode (normal operation)
2	defrosting mode
<i>a</i>	air
<i>amb</i>	ambient
<i>aw</i>	air-to-water
<i>b</i>	bubble point (refrigerant data)
<i>build</i>	ice building (at the evaporator)
<i>br</i>	brine
<i>c</i>	condenser / condensation
<i>c</i>	critical point (refrigerant data)
<i>cmp</i>	compressor
<i>cyc</i>	cycle
<i>d</i>	dew point (refrigerant data)
<i>defrost(ing)</i>	in defrosting mode
<i>e</i>	evaporator / evaporation
<i>ec</i>	evaporator-to-condenser
<i>enfdefr</i>	enforced defrosting (of the evaporator)
<i>exp</i>	expansion valve
<i>f</i>	floor (building model)
<i>fan</i>	air fan

<i>g</i>	gas phase
<i>he</i>	internal heat exchanger
<i>hg</i>	hot gas
<i>hgc</i>	hot gas cooling
<i>hgv</i>	hot-gas valve
<i>HP</i>	(overall) heat pump
<i>hp</i>	high pressure
<i>i</i>	index
<i>i</i>	at the input/inlet
<i>ice</i>	icing process (at the evaporator)
<i>l</i>	liquid phase
<i>limit</i>	limit value
<i>lp</i>	low pressure
<i>lv</i>	liquid valve
<i>m</i>	mean value / mid-point (refrigerant data)
<i>mass</i>	ice mass (at the evaporator)
<i>max, min</i>	maximum, minimum
<i>mode</i>	operating mode
<i>nom</i>	nominal (fault-free)
<i>o</i>	at the output/outlet
<i>off</i>	in shut-down mode
<i>on</i>	in operational mode
<i>ov</i>	superheating
<i>pump</i>	(water) circulation pump
<i>r</i>	refrigerant
<i>r</i>	room (building model)
<i>ref</i>	reference (parameter)
<i>s</i>	heat source
<i>s, sa</i>	saturation
<i>sc</i>	subcooling
<i>sc</i>	scaled value
<i>seq</i>	(data) sequence
<i>selfdefr</i>	self-defrosting (of the evaporator)
<i>sp</i>	setpoint
<i>ss</i>	in steady-state
<i>step</i>	step (input)
<i>sw</i>	source-to-water
<i>tot</i>	total (extent)
<i>v</i>	vapour phase
<i>w</i>	water / heat sink

## 2 Introduction

### 2.1 Motivation

Regarding ecological and economical aspects, the main issue is to maximize the efficiency of the refrigerating machines, air-conditioners, and heat pumps, in order to minimize their energy consumption. Minimizing the global energy consumption and CO<sub>2</sub> emission is mandatory for mitigating global warming. For heat pumps, the COP (coefficient of performance) is to be maximized. In particular, any degradation of the performance during operation should be detected early and the reasons for the degradation should be diagnosed.

A field analysis of Swiss heat pump installations showed that the averaged energetic efficiencies of the heat pumps installed in the buildings were about 10% lower than the efficiencies measured under standard laboratory conditions ([Erb 00], cf. Section 2.2). For some installations, the performance degradation was as high as 30%. Since the reasons for these degradations vary from one installation to another, a diagnosis system is required, which is able to monitor a particular installation at start-up and during operation. Once the reasons for the degradation are known, an optimization of the particular installation or of future installations is possible.

Therefore the project ‘Short-Term Rating Method’ of the Swiss Federal Office of Energy (cf. Section 2.2) was started. The purpose of the project is to develop a fault detection and diagnosis method for operational monitoring of a heat pump heating system. Within this thesis, the heat pump as a subsystem is considered. For a diagnosis system, the following requirements have to be fulfilled:

- The costs for the additional hardware and software must be minimal.
- The number of sensors required must be as low as possible. Temperature and/or pressure sensors are preferred, but no heat flow or mass flow sensors are to be used.
- The maximum number of faults are to be diagnosed. Especially, the faults described in Section 4.2 have to be recognized.
- The diagnosis quality has to be sufficient in order to have a high reliability. Especially no false alarms should occur.
- A fast commercial realization with a low implementational effort should be possible.

With a diagnosis system, the condition of the machine and the size of potential faults is known at all times. Fig. 1 shows the progression of a fault from its initial size at startup to its maximum size, when the machine fails or its functionality is strongly impaired.

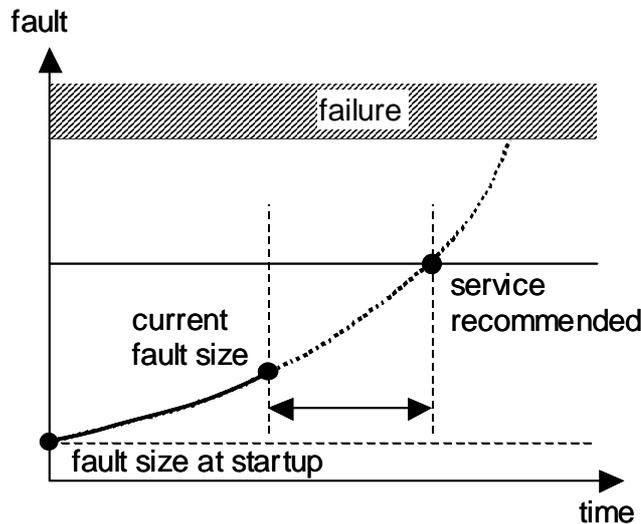


Fig. 1: Progression of a fault over the time.

The faults are diagnosed both at startup and during the operation. At startup, any fault indicates a deviation of the machine condition from design. With this knowledge, the system can be optimized, if necessary.

Another scope of the diagnosis during operation is the improvement of the maintenance strategy. Three different strategies for maintenance are possible [Kuhlmann 00], [Neuenssch. 96]:

- *failure-oriented* – The machine is running until it fails or a severe alarm is caused by a temperature or pressure signal, which exceeds an alarm limit (for example, high pressure or low pressure alarms). Machine failure generates costly repairs, accompanied by downtime. Additionally, the efficiency of the machine decreases for a long time before failure, which is not detected by the alarm limits.
- *time-oriented* – This strategy is based on service intervals, which are scheduled in advance. In order to prevent failure, the time intervals tend to be too short, since an exact prediction of the lifetime is very difficult for any particular machine. Hence, the redundant inspections and services cause costs that could be saved. In this strategy as well, the efficiency of the machine might decrease before inspection.
- *condition-oriented* – With the fault diagnosis systems presented in this thesis, the current condition of the machine is always known. An in-

crease of a fault is recognized early. Using this knowledge, it can be decided whether a service is recommended or not (cf. Fig. 1). Optionally, the remaining time to the next service can be estimated by looking at the fault trend charts. With this idea, the operational costs as well as the service costs can be reduced [Birchler 01].

The fault diagnosis systems presented in this work are not only applicable to heat pumps, but also to larger heating, refrigerating, or air-conditioning systems. For large-scale plants, the potential savings would even be higher due to an optimal operation and a condition-oriented maintenance.

## 2.2 The Heat Pump Market

An international analysis of operating heat pump systems is presented in [IEA-HP 01/1]. ‘Heat pumping’, i.e., reversing the natural heat flow from a higher to a lower temperature level, is used in many applications for producing cold (refrigerators, air-conditioners, large refrigerating machines), heat (in residential or industrial heat pumps) or both (in reversible air-conditioner systems or heat pump systems with integrated cooling system).

Worldwide, about 120 million ‘heat pumping systems’ are in operation, mainly in Japan and the United States. The efficiency of about 60% of the systems sold in the United States still is no higher than the minimum allowable, which is at  $SPF > 1.99$  for heating ( $SPF =$  seasonal performance factor = mean energetic efficiency of a heating season). However, the efficiencies are slowly rising due to improvements. About 1.5 million heating-only systems are operating in Europe. Mainly in Sweden, the Netherlands, Austria, Germany and Switzerland, efforts are being made to improve the efficiency of such systems.

For Switzerland, an overview of the heat pump market is given in [FWS 99]. About 60’000 heat pumps have been installed so far in heating systems. Every year, about 6’000 new heat pumps are installed. About one third of the new residential buildings are equipped with a heat pump heating system. Mainly ground sourced (brine-to-water) heat pumps and air sourced (air-to-water) heat pumps are used. The results of the field analysis of the Swiss heat pump installations are shown in [Erb 00] and [IEA-HP 01/1]. Therefore, measurements of about 120 installations have been statistically analysed. During the period 1994 to 1998, the overall  $SPF$  of those systems increased from 2.6 to 3.2.

For controlling the quality of the heat pumps, the Heat Pump Test Centre at Töss (Switzerland) measures the efficiencies under standard laboratory conditions. Today, the mean COP of the heat pumps tested is about 3.5, which is almost 10% higher than the overall SPF value measured in the field (COP = coefficient of performance = ratio of heating power output to electrical power consumption at given temperature levels). For some installations, the performance degradation in the field was even higher, up to 30% [Erb 00].

In [Erb 00], there are speculations on the reasons for these deviations, but it is difficult to find general explanations because of the high number of potential problems which vary from one installation to the other. The scope of the project ‘short-term rating method’ therefore was to develop a diagnosis system able to explain those deviations for the particular installations.

### 2.3 The Project ‘Short-Term Rating Method’

The project ‘short-term rating method’, financed by the Swiss Federal Office of Energy, was divided into the phases 1 to 5. This thesis has been developed within the phases 4 and 5.

**Project phases 1 to 3.** During the phases 1 to 3 of the project, data of a residential building at Barzheim (Schaffhausen, Switzerland) were collected and analyzed with conventional (manual) methods. As a result of that analysis and of long-time experiences at many other objects, the most relevant fault cases of heat pump heating systems in residential buildings were summarized in a fault-tree [Reiner, Shafai 98]. The faults were separated into those that occurred in the heat distribution and emission system, the building or the control system (external faults), and those at the heat pump or its interface (internal faults). A list of potential internal faults is given in Section 4.2. Models of the heat load, consisting of the heat distribution and emission system and the building, were developed in the project phases 1 to 3 as well [Reiner, Shafai 98].

**Project phases 4 and 5.** For this thesis, only the heat pump as a subsystem is considered and a simulation model as well as two fault diagnosis systems are developed ([Zogg 01/2], [Zogg 99]). For testing the diagnosis systems with real data of various fault cases, an existing test bench at the University of Applied Sciences at Winterthur (ZHAW) was used, and a new test bench was built at the Measurement and Control Laboratory (IMRT) of the Swiss Federal Institute of Technology at Zurich (ETH).

**Future project phases.** The next step will be the commercial realization of the diagnosis systems (cf. outlook in Chapter 14). Therefore, the appropriate methods have to be chosen and implemented for a particular installation. Some suggestions for the realization process are made in Section 14.1.

## 2.4 Separation to other Projects

Experiences from operating heat pump systems are summarized in [IEA-HP 01/1] for the international market, and in [Reiner, Shafai 98] for some installations in the Swiss market. In Europe, and particularly in Switzerland, hydronic heat distribution systems dominate the market. Often, these systems cause problems because of wrong planning or unskilled installers. In order to reduce the high number of different systems, standard schemes are developed within the project described in [Afjei 01]. These schemes will be used as references. Typical planning mistakes result in heat pumps that are too large, which then work at a low part-load efficiency (with frequent on-off switching). Such planning faults have to be eliminated before installing the heat pump and are not the main focus of the diagnosis systems presented in this thesis. Often, the settings of the heat pump controllers are wrong, which results in too high supply temperatures on the water side or in frequent on-off switching. Better control strategies based on predictive optimization are suggested in [Wimmer 01]. The relationship between the switching frequency and the degradation of performance because of the start-up losses was investigated in [Gubser 99]. Because of the icing effect in air-to-water heat pumps, the evaporator has to be defrosted periodically. During the defrosting intervals, the heating power is used for melting the ice layer and therefore is lost. Many known control strategies for defrosting are summarized in [Hubacher 00] where some suggestions are made for improvements. One of the suggested strategies is based on the diagnosis systems presented here (cf. Section 9.1 and Chapter 14).

## 2.5 Chronology of Development

This thesis presents various models and methods for fault diagnosis. This section shows in which order and why they were developed. Note that the order of the chapters in this thesis differs from the chronological order of development. The chapter overview is given in Section 2.6, where the chapters are ordered by their importance subject to fault diagnosis for heat pump systems. The chronology of development is treated below.

First, a detailed physical and dynamic model of a heat pump was developed in order to become acquainted with the subject of research (cf. Chapter 11). After a high modeling effort, it was possible to validate the model by measured data for one particular heat pump. The simulated signals are not exactly identical to the measured signals, but the model is able to represent the significant characteristics and dynamics of the heat pump. The first idea was to use this model approach for fault diagnosis as well. But it appeared to be much too complex for a later on-line application. For understanding the diagnosis systems, the knowledge of Chapter 11 is not urgent. Thus it is placed after the presentation of the diagnosis systems. Nevertheless, it is useful for those readers who wish a physical insight into the functionality of a heat pump and to the effects of the faults. The detailed simulation model has also been used to train and test the diagnosis systems (see below).

The model approach had to be simplified, which led to a steady-state approach with a significantly reduced set of equations. The physical steady-state model approach was then directly used for a diagnosis system with steady-state parameter identification, called *HeatWatch* (cf. Chapter 8). This diagnosis system calculates and displays the physical parameters of the monitored heat pump, but does not interpret them.

The number of required sensors is given by the physical equations in steady-state, which results in a rather high number of sensors for a given number of parameters. This statement can be explained by a simple example of a heat exchanger, which is used as evaporator or condenser in a heat pump. For calculating one parameter, namely the heat transfer coefficient between the warm and cold side, four temperatures at the inputs and outputs of both sides, as well as the mass flow at one side has to be measured, which adds up to five sensors. In a similar way, the number of sensors is given for an entire heat pump system.

In the next step, another diagnosis system was developed, called *FuzzyWatch* (cf. Chapter 5), which implies a method to reduce the number of sensors without using physical models in steady-state or even more complex physical models. The aim was to develop an algorithm which automatically finds the minimum number of signals and automatically generates an appropriate model for a given set of measured data. This algorithm is intended for data sets of all types of heat pumps or refrigerating machines without requiring an extra modeling effort. Therefore, a statistical approach with a gray-box model structure was chosen. The structure of the model is given by physical considerations, but its parameters have no physical meanings. Thus, the parameters could not directly be used as diagnosis results. The diagnosis system had to be extended by a classification step, which associates the parameters to the corresponding faults. As an advantage over *HeatWatch*, here the interpretation of the parameters is automated and the faults are directly displayed.

*FuzzyWatch* needs to learn the classification of the faults during a training phase. Therefore, measured data is fed to the system, together with the corresponding fault cases. The data is either measured on a test bench or generated by simulation. First, an existing test bench (ZHW, cf. Section 2.6) was used to measure data of the most important fault cases. Later, a new test bench was built for a commercial heat pump (IMRT, cf. Section 2.6). For testing the fault cases, the commercial heat pump was modified. The data of the test benches was not only used for training, but also for testing the trained system (with different data).

The detailed physical heat pump model was used for training and testing *FuzzyWatch*. Thus, the simulation model was regarded as a ‘virtual’ test bench, with which a large amount of data for a high number of fault cases was generated without any experimental effort. The first idea was to use the simulation model generally in future applications (for new heat pump types or refrigerating machines) in order to replace the experimental training on a test bench. But the modeling effort is high for an *exact* adaptation to a new machine type. This effort has to be weighted up with the experimental effort of the training on a test bench. Moreover, the simulation model can only be validated for the nominal (fault-free) case, because for validating the fault cases, a test bench would be required for generating data of the fault cases.

Nevertheless, the simulations were useful for establishing a link between the statistical (gray-box or even black-box) approaches of *FuzzyWatch* and the physical model. Furthermore, the diagnosis systems were tested on many different data sets from the residential building, two test benches, and the simulation model.

## 2.6 Overview

Two diagnosis systems were developed. As a first diagnosis system, *Fuzzy-Watch* is presented, which implies fault classification and only requires a low number of sensors. The second diagnosis system, *HeatWatch*, emphasizes a fast commercial realization because of a simple method and a very small training effort.

***FuzzyWatch.*** The fault diagnosis system described in Chapter 5 first identifies the parameters of a dynamic model and then classifies the faults from the parameters. As a result, the actual fault codes are displayed. With an automatic algorithm, the diagnosis system can be adapted to any heat pump type or any type of refrigerating machine. Before operation, it needs to be trained by experimental data for known fault cases measured on a test bench. During the fully automated training phase, the internally identified parameters are associated to the fault cases and the number of sensors required is minimized. For online operation, the trained system is then used to classify the current data sequences. Therefore, the parameters are sequentially re-identified, using a least-squares algorithm. For the classification of the faults, statistical approaches as well as special methods from fuzzy logic and neural network theory are used. The methods have been simplified in order to reduce the computational effort for training and online operation.

***HeatWatch.*** The fault diagnosis system described in Chapter 8 identifies the physical parameters and calculates the characteristics of the machine, using a steady-state model. Analyzing these parameters and characteristics, the faults are interpreted by the service technician or the user, but they are not classified automatically. Before operation, the diagnosis system needs to be initialized by nominal (fault-free) data, but no data of any fault case is required. During online operation, the parameters are sequentially re-identified in steady-state, using a matrix inversion.

**Tests.** Both fault diagnosis systems have been tested by data from different applications (cf. Table 1). For a first test of the diagnosis system *HeatWatch*, measured data of the residential building at Barzheim (Schaffhausen, Switzerland) with a standard commercial air-to-water heat pump is used (Application I). By analyzing the data of Barzheim, mainly external faults were found, such as malfunctions in the heat distribution system or wrong controller settings, partly due to wrong planning. Regarding the heat pump only, this data is considered as nominal data for the ‘fault-free’ case, because no internal faults occurred. Nevertheless, some interesting effects such as icing of the evaporator were observed in the identified parameters. The diagnosis system

*FuzzyWatch* first was trained for the data of a brine-to-water laboratory heat pump at the University of Applied Sciences at Winterthur (ZHW, application II). With this test bench, data was acquired for several fault cases. Based thereon, various clustering methods were compared (cf. Section 5.4).

**Simulation Model.** A next data set was generated with a detailed, physical heat pump simulation model, which was developed within the scope of this work (cf. Chapter 11). With the simulation model, data for many fault cases, gradual faults, and combinations of simultaneous faults were generated (application III). The diagnosis system *FuzzyWatch* was tested for training data as well as for validation data. The simulation model is especially designed for testing the diagnosis system without any experimental effort. It was validated with the measured data from Barzheim.

*Table 1: The fault diagnosis systems and their applications, chapter overview, and main focus of each chapter*

	Heat Pump Type	<i>FuzzyWatch</i>	<i>HeatWatch</i>
Idea, Theory		Chapter 5	Chapter 8
Application I: Barzheim data	air-to-water commercial (standard)		Ch. 9, Section 9.1 data from real building (nominal)
Application II: ZHW test bench	brine-to-water laboratory	Ch. 6, Section 6.1 comparing various clustering methods	
Application III: Simulation data	air-to-water commercial (simulated)	Section 6.2 high number of fault cases	
Application IV: IMRT test bench	brine-to-water commercial (modified)	Section 6.3 on-line diagnosis	Section 9.2 on-line diagnosis
Software for IMRT test bench	brine-to-water commercial (modified)	Chapter 7 on-line diagnosis	Chapter 10 on-line diagnosis

**Test Bench.** Both diagnosis systems were tested on a further test bench (application IV), which was built within the scope of this work at the Measurement and Control Laboratory (IMRT) of the Swiss Federal Institute of Technology, Zurich (cf. Chapter 12 and Fig. 2). This test bench contains a commercial brine-to-water heat pump, which was modified and equipped with additional sensors in order to generate data for several fault cases. With this data, especially the on-line versions of both diagnosis systems were tested.

**Software Tools.** For each diagnosis system, a software tool with a graphical user interface was developed. Here, one version of each software tool will be presented, adapted to the data of the IMRT test bench. In these software tools, the methods of Chapters 5 and 8 are implemented. They are considered as examples to show how the online diagnosis results can be visualized. The training procedure (for *FuzzyWatch*) as well as the initialization procedure (for *HeatWatch*) are also integrated.



*Fig. 2: The test bench at the Measurement and Control Laboratory (IMRT).*



## 3 An Introduction to Fault Detection and Diagnosis

### 3.1 Definitions

Generally, two steps are distinguished, namely fault detection and fault diagnosis [Gertler 98].

**Fault detection.** The first step indicates that something is going wrong in the monitored system, which means that a fault is present.

**Fault diagnosis.** In the second step, the fault is described in detail. It consists of the two parts fault isolation and identification.

- *Fault isolation:* The determination of the exact location of the fault.
- *Fault identification:* The determination of the magnitude of the fault.

The fault diagnosis systems presented in this work combine fault detection and diagnosis in one step. They directly diagnose the location (and the magnitude) of the current fault, and therefore the detection is automatically included. For the user interface of the software tools, a switching between the detection mode and the diagnosis mode is useful (cf. Chapters 7 and 10).

### 3.2 Methods

A fault diagnosis system may consist of the following modules (some overviews are also given in [Frank 96], [Frank 00]).

**Signal Acquisition and Limit Checking.** The signals of the plant are measured. They can directly be used for fault diagnosis (model-free methods). The most common method is limit checking of signals with thresholds. Since the signals are not only sensitive to the faults, but also to changing inputs or operating conditions, the thresholds need to be conservative (i.e. sufficiently high). Therefore, only large faults cause an alarm such that in an early stage no diagnosis is possible. Additionally, one fault may propagate to many variables and set off a confusing multitude of alarms, making fault isolation very difficult.

**Feature Generation.** Typical features have to be extracted from the signals. They can be represented by signal parameters in the time domain, in the frequency domain, or by statistical properties. But they can also be represented by model parameters. This is the approach used in this work (model-based

methods with parameter identification, [Isermann 96], [Fuente 96], [Zogg 98]). In general, the model parameters are much less sensitive to changing inputs or operating conditions than the signals. The heat pumps, for which the diagnosis systems are designed in this work, are controlled by on-off switching. Thus, the data sequences are characterized by step responses. One data sequence is described by one parameter set, which is a significant data reduction.

**Symptom Generation.** Symptoms are the differences between the features in the case of a fault and the features in the nominal (fault-free) case. Here, the parameter deviations from their nominal values are regarded as fault symptoms. The diagnosis system of Chapter 8 displays the deviations of the physical parameters. Because of the physical meaning of the parameters, the faults can then be easily deduced by the service technician. On the other hand, the diagnosis system of Chapter 5 is based on gray-box or black-box parameters, which then are classified by the next step (see ‘Fault Classification’ below).

**Fault Classification.** From the symptoms, the causative faults have to be found. Therefore, fault trees can be used as well as logical reasoning (for example expert systems). Other approaches are based on “pattern recognition” techniques, which compare the actual patterns in the symptoms (or features) to known patterns for different fault cases [Stylianou 96]. For all approaches, some knowledge about the faults is needed. The knowledge can either be a user input to the system (for example to an expert system), or it can be learned by the system during a training phase. Statistical techniques [Rossi 97] as well as techniques based on fuzzy logic and neural networks [Patton 00] are used for this purpose. In this thesis, some special methods are extracted from these techniques, which then are modified and combined in order to handle the heat pump applications. During the training phase, methods from fuzzy logic are used to build clusters in the parameter space (derived from [Höppner 97], [Chiu 94]), which then are used to build special types of neural networks with statistical properties (similar to radial basis function networks in [Koivo 94], [Roverso 00]).

**Residuals.** In literature, the symptoms are often defined by residuals, i.e., the differences between the measured signals and the signals generated by nominal models [Rossi 97], [Isermann 96], [Gertler 98]. Related methods are based on observers, Kalman filters, or other techniques for generating residuals [Blanke 95], even using fuzzy relational models [Amann 01]. These model-based approaches are related to the parameter identification approach. This relation is described in Appendix B. However, with these residuals, the advantage of the significant data reduction would be lost, because the residuals are data sequences of the same length as the signals. Additionally, most of the faults in the heat pump (and in many other applications) are parametric (multiplicative), which leads to the conclusion that parameter identification is the natural approach.

**Principal Component Analysis (PCA).** This approach is used when a high amount of data from many sensors has to be evaluated in order to reduce the data space [Milek 99]. In this thesis the ‘maximum information’ should be extracted from the data of a minimum number of sensors. Here, the data is reduced by the parameter identification approach.

### 3.3 State-of-the-Art Applications to HVAC Systems

An overview of numerous fault detection and diagnosis methods that have been applied to heating, ventilating and air-conditioning (HVAC) systems is given in [IEA 96]. Some theses relevant to the present research are discussed below. For a better understanding of the relations between the methods, Chapters 5 and 8 should be read as well.

Some diagnosis methods for heat exchangers are found in [Kuhlmann 00], [Thomson 00], [Weyer 00], and others, where the heat transfer parameters are calculated from the signals. The same method is used in [Neuenssch. 96], where a very detailed model of a heat exchanger has been developed. From the change of the heat transfer parameter  $k$ , fouling can be diagnosed, which is also described by the ‘fouling resistance’  $R_f$ . The calculation of the heat transfer parameters  $k$  for the evaporator and condenser is also part of the diagnosis system *HeatWatch* (cf. Chapter 8). But in contrast to [Neuenssch. 96], no mass flows are measured here. In addition to the heat transfer parameters, numerous other physical parameters are calculated for describing all modules of the entire heat pump system.

A method based on signal parameters (using a signal model) is presented in [Räber 97]. In that work, one characteristic temperature signal is converted to its spectrum. The features are extracted from the spectrum at several discrete frequencies. As a result of the training phase, each fault case is represented by a point (location) in the feature space. During operation, the faults are then classified by calculating the (minimum) distances between the current point in the spectral feature space and the points of the trained fault cases. That idea is similar to the clustering techniques used for the diagnosis system *FuzzyWatch* (cf. Chapter 5). Here, the spectral feature space is replaced by the parameter space, the points of the faults in the feature space are replaced by clusters with a distribution function, and the calculation of the distances is replaced by the calculation of the membership grades. But as a fundamental difference, here input-output models are used instead of signal models, which in general are less sensitive to changing inputs or operating conditions.

Logical reasoning for fault classification is used in [Maurer 92], [Maurer 96], which is based on an expert system. The rules are inputs from an experienced expert. Many rules have to be defined with an exact description of the input variables, which in general can be rather sophisticated in order to yield reliable results and to prevent contradictions. Signals or signal parameters as well as model parameters can be used as input variables. Fault-trees, which describe the association between the faults and the symptoms (signal or parameter deviations) can also be expressed in rules and integrated into expert systems. A fault-tree has been defined in [Reiner, Shafai 98] for an entire heating system in a residential building. The fault list in Table 2 on page 41 has been extracted from that fault tree. The diagnosis system *HeatWatch* identifies the parameters in that list. Since the parameters have a physical meaning, the interpretation can be done by the service technician or the user. Thus, no expert system is required. The diagnosis system *FuzzyWatch* includes the fault classification as a further step. There the rules are learnt automatically during the training phase (cf. fuzzy rules in Section 5.4.1) such that no input from an expert is required.

## 4 The Heat Pump

The refrigerant is used as a working fluid in a heat pump or a refrigerating machine in order to transport the heat from the heat source to the heat sink. Today, HFCs (hydrofluorocarbons) are used as refrigerants, such as R407C, which is used in the commercial heat pumps investigated in this thesis. They replace the former used CFCs (chlorofluorocarbons) and HCFCs (hydrochlorofluorocarbons), which impact ozone depletion. But the HFCs still have a high global warming potential [IEA-HP 01/3]. Thus, leakage of the refrigerant during operation should be minimized. In future the HFCs will have to be replaced by other refrigerants, such as HCs (hydrocarbons), CO<sub>2</sub>, or ammonia.

Global warming is not only due to refrigerant leakage (and disposal). A great part of the global warming impact is due to the CO<sub>2</sub> emissions, which are related to the energy consumption of the heat pump. Therefore, a high COP (coefficient of performance) and thus a low energy consumption is urgent.

### 4.1 The Refrigerant Cycle

Generally the machine contains an evaporator, a compressor, a condenser, and an expansion valve (cf. Fig. 3). For heat pumps the heat source medium may be air, water, or brine in the case of a ground-sourced heat pump. The brine medium consists of water with about 25%...35% of antifreeze (ethylene glycol), which circulates in a ground cycle. On the other hand, the heat sink medium is air or water. The water circulates in the heat distribution system of the building.

Fig. 4 shows the temperature-entropy-diagram of the internal refrigerant cycle [Baehr 92]. In the evaporator, the refrigerant is evaporated at low pressure  $p_{lp}$  and at low temperature ( $4 \rightarrow 1'$ ) and superheated ( $1' \rightarrow 1$ ). The refrigerant is then compressed to high pressure  $p_{hp}$  ( $1 \rightarrow 2$ ). An ideal compressor would reach point 2\* without any change in the entropy  $s$ , but a real compressor reaches point 2 because of dissipation, and therefore needs more power for the same pressure rise. In the condenser the hot gas is cooled down ( $2 \rightarrow 2'$ ), the refrigerant is condensed at high pressure and high temperature  $p_{hp}$  ( $2' \rightarrow 3'$ ), and subcooled ( $3' \rightarrow 3$ ). During the expansion the enthalpy  $h_{exp}$  is kept constant, but the entropy is increases ( $3 \rightarrow 4$ ). At point 4, both the liquid phase as well as the vapour phase are present with a ratio of  $x_{e,i}$ .



The thermostatic expansion valve (TXV) controls the superheating temperature difference  $\Delta T_{ov}$  ( $T_1 - T_1'$ ). A minimum amount of superheating is needed for the proper operation of the compressor, because the liquid phase should not reach the suction line. The mechanically operated valve works like a P-controller, which feeds back the superheating  $\Delta T_{ov}$  and adjusts the mass flow  $m_{exp}^*$  to the evaporator (cf. Fig. 5, left side). The superheating sensor is realized by a capillary filled with refrigerant. This sensor is attached to the evaporator output tube, and feeds the pressure signal  $p_{ov}(T_{ov})$  corresponding to the superheating temperature  $T_{ov}$  to the valve. Thus, the valve opening  $u_{exp}$  is set by the pressure difference between  $p_{ov}(T_{ov})$  and  $p_{lp}$  at the input of the evaporator, which works against a spring. With the characteristics of the expansion valve and the characteristics of the superheating process, the operating point is determined (cf. Fig. 5, right side).

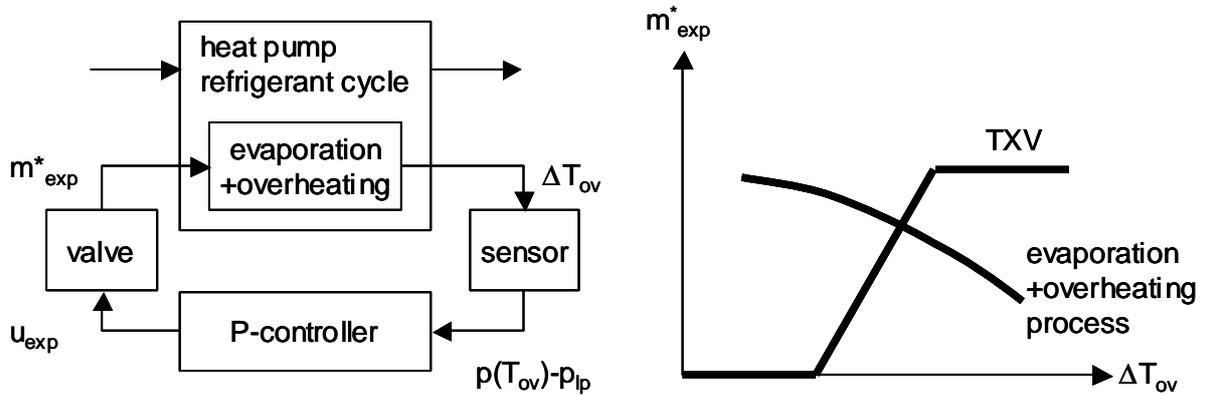


Fig. 5: The thermostatic expansion valve (TXV). Left side: internal control loop. Right side: characteristics of the expansion valve (mass flow  $m_{exp}^*$ , superheating  $\Delta T_{ov}$ )

Due to a large gain of the P-controller in combination with the sensor delay and the saturation of the valve opening  $u_{exp}$ , the thermostatic expansion valve often oscillates during operation. This ‘hunting effect’ is described in [Gruhle 87]. Apparently, these oscillations do not significantly decrease the lifetime of the expansion valve or the heat pump. But the measured signals are superposed by oscillations, which are more or less strong for signals at different locations. In larger applications, electronic expansion valves are used instead of mechanical expansion valves, in order to improve the control quality and to eliminate the oscillations.

Fig. 6 shows the external input and output signals of a heat pump, such as the water feed flow temperature  $T_{w,o}=T_6$ , the water return flow temperature  $T_{w,i}=T_8$ , and the water mass flow  $m_w^*$  at the heat sink side (cf. Fig. 3). Most of the modern heat pump heating systems work with floor heating systems at a low water temperature level, which results in a higher COP. In residential

buildings, on/off-controllers are used most often. They measure the water return flow temperature  $T_{w,i}$  and the outdoor temperature. On the heat source side, the external signals are represented by the feed flow temperature  $T_{s,o}=T_5$ , the return flow temperature  $T_{s,i}=T_7$ , and the heat source mass flow  $m_s^*$ .

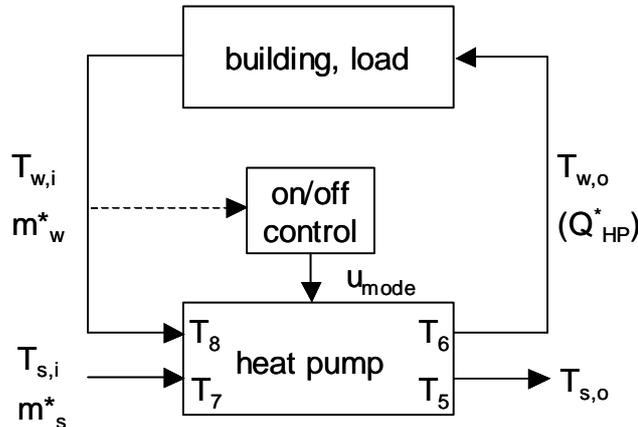


Fig. 6: The integration of a heat pump into a building system. The heat pump is controlled by an on/off controller.

## 4.2 Potential Faults

In heat pump systems, several fault cases may cause an abnormal operation and reduce the performance. Relevant fault cases of real systems have been investigated and summarized in [Reiner, Shafai 98], which led to the list in Table 2.

During operation, fouling might take place in the heat exchangers ([VDI 91], [Bott 95]). In general, these are slowly increasing (gradual) faults. Fouling of the evaporator reduces its heat transfer parameter (first row of Table 2). During operation, soiling of the heat source cycle results in a decreasing source mass flow (second row). At startup or during operation, a malfunction of the circulation pump on the heat source side or of the air fan also results in a reduced (or zero) source mass flow, which is a sudden fault (second row).

In the case of an air-to-water heat pump, an ice layer might form at the air side of the evaporator if the air temperature is low and the air humidity is high. This effect is not treated as a real fault, but it also reduces the heat transfer parameter at the evaporator (first row). Therefore, air-to-water heat pumps need a defrosting cycle, which is activated periodically (for example with intervals of about one hour) for melting the ice layer.

Heat exchanger fouling might also affect the condenser (third row). Especially in floor heating systems with a low water temperature level, the water cycle (including the heat distribution and emission system as well as the condenser) might silt up increasingly during operation, which reduces the condenser heat transfer parameter as well as the water mass flow through the condenser (third and fourth rows). At startup or during operation, a malfunction of the water pump results in a reduced (or zero) water mass flow, which is a sudden fault (fourth row).

Table 2: Potential faults and changed parameters.

changed parameter	faults (in brackets: for air-to-water heat pump)
evaporator heat transfer parameter	fouling of the evaporator on the heat source side (or icing of the evaporator on the air side) fouling of the evaporator on the refrigerant side
heat source mass flow	soiling of the heat source cycle (or icing of the air channel) source pump malfunction (or air fan malfunction)
condenser heat transfer parameter	silting-up of the condenser on the water side, especially in combination with floor heating systems fouling of the condenser on the refrigerant side
water mass flow	silting-up of the water cycle, especially in combination with floor heating systems water pump malfunction
characteristics of the compressor	malfunction of the compressor
characteristics of the expansion valve	malfunction of the thermostatic expansion valve
refrigerant filling	refrigerant leakage
COP or exergetic efficiency	used for fault detection: influenced by all faults above

A compressor malfunction such as internal leakage results in a change of the compressor characteristics (fifth row), which goes together with a lower compressor efficiency and a lower refrigerant mass flow. The expansion valve might be partially blocked or its functionality might be impaired (sixth row). Thus its characteristics might change and the quality of the superheating control might be reduced. Malfunction of the compressor or expansion valve are either sudden or slowly increasing (gradual) faults.

Refrigerant leakage generally occurs slowly during operation (eighth row). As a result, the refrigerant filling of the heat pump decreases. The COP or exergetic efficiency (cf. Section 8.4) are (more or less) influenced by all faults above, and are used for fault detection (decision, if something is going wrong, cf. Section 3.1). As discussed at the beginning of Chapter 4, the main criterion is a high COP or exergetic efficiency during operation in order to minimize the energy consumption. By knowing which faults of Table 2 are present, the reasons for a degradation in performance are known, as well. Refrigerant leakage is treated as a special fault, which should be low because of its additional high global warming potential.

The fault diagnosis systems presented in this thesis generally diagnose the faults in the list of Table 2. The fault list is adapted to each application, depending on the type of heat pump referred and the number of fault cases investigated. The diagnosis system *FuzzyWatch* (cf. Chapter 5 and applications in Chapter 6), directly diagnoses the faults in the right-hand column of Table 2, whereas the diagnosis system *HeatWatch* (cf. Chapter 8 and applications in Chapter 9) calculates the parameters in the left-hand column, from which the faults can be interpreted.

## 5 A Fault Diagnosis System with Integrated Fault Classification – Theory

This diagnosis system integrates parameter identification as well as fault classification (Fig. 7). The signals are measured by temperature sensors, whereas the number of required sensors depends on the ‘information content’ of the signals. Only those signals are selected, that are best suited for diagnosis.

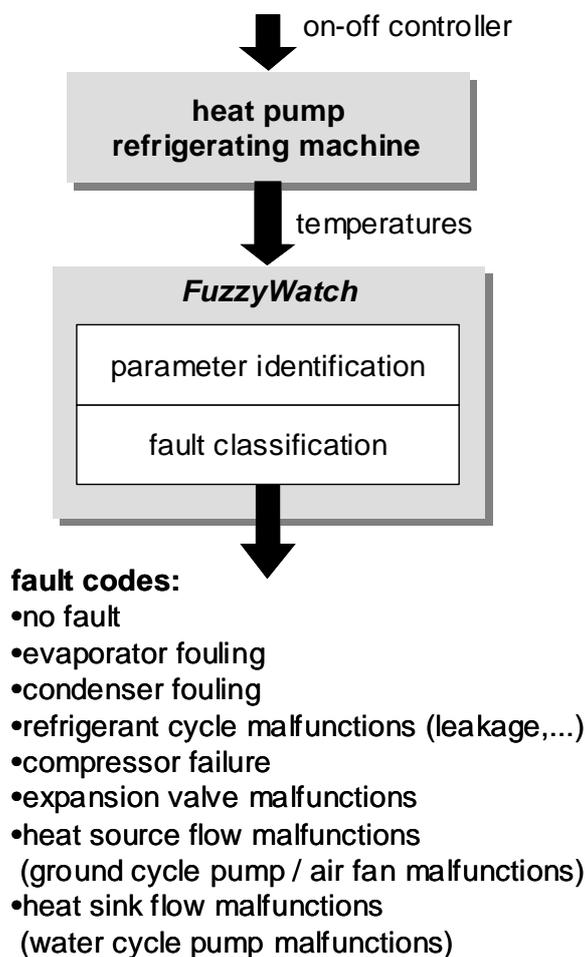


Fig. 7: The schematic representation of the fault diagnosis system called FuzzyWatch.

### 5.1 Concept

The fault diagnosis system presented in this chapter first has to be trained by experimental data from a test bench or by simulation data. During the training phase, the fault diagnosis system learns the classification of the faults

from the parameters identified using the measured signals of the nominal case as well as the fault cases (Fig. 8, on the left and Section 5.1.1). The trained system is then used for fault diagnosis during the operational phase (Fig. 8, on the right and Section 5.1.2).

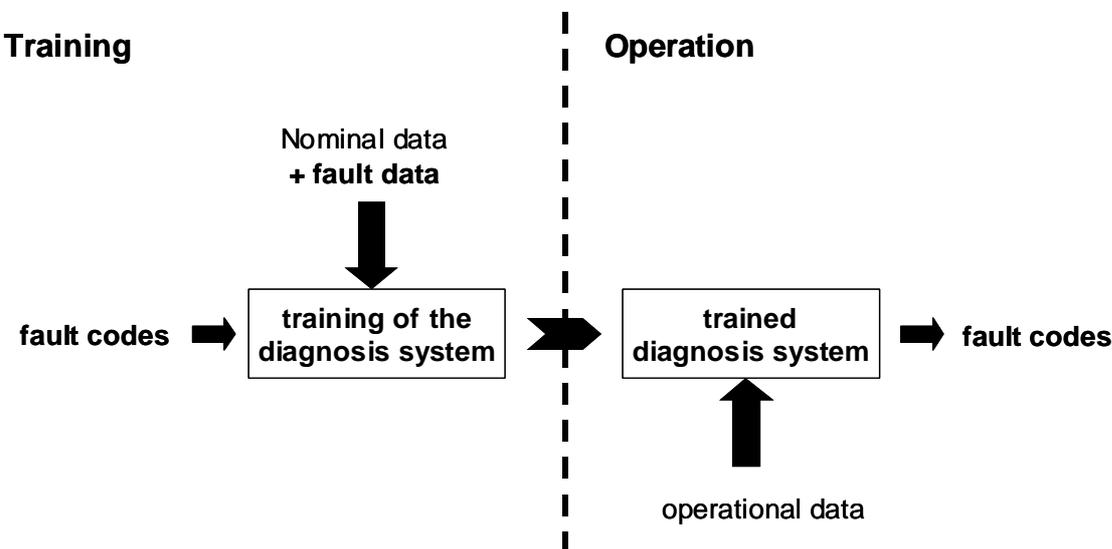


Fig. 8: Training and Operation.

### 5.1.1 Training Procedure

The training procedure is designed in such a way that it can be automated (cf. software tool in Section 7). As an input, measured (or simulated) data sets are needed for all fault cases that are to be trained. For any particular application, a fault classification system is automatically generated therefrom. The procedure is generally applicable to the measured data sets of all types of heat pumps or even of other machines, such as refrigerating machines or air conditioners. In this work, it is successfully applied to three data sets of different heat pump types (cf. Chapter 6). Thus the main focus is directed toward a general procedure, which provides several methods for modeling and learning that can be used for a wide range of applications. For a particular application, the variety of the methods could be reduced, or manual optimizations could be made, such as using more specialized (detailed) model approaches or modifying the learning algorithms. But since a specialization in one application would result in a lack of generality, specializations are not the subject of this thesis.

In order to generate training data sets, some test cycles are run for each fault case (cf. Fig. 9). During one test cycle  $i_{cyc}$ , the heat pump is switched on and off several times, which results in several data sequences  $i_{seq}$ . Each data se-

quence is evaluated individually. For the training on the IMRT test bench, the duration of the off-on sequences is set to a minimum of 10 minutes (cf. Section 12.3.1 for details).

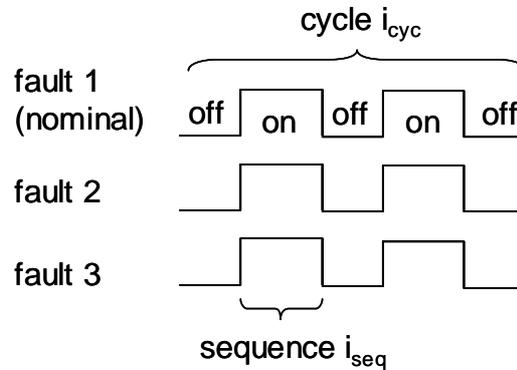


Fig. 9: Test cycles for training

An output signal  $y_i$  for one off-on sequence is illustrated in Fig. 10, which might be a temperature signal  $T_1..T_6$  from a sensor located in the refrigerant cycle, or at the heat source or heat sink side (cf. Fig. 3 on page 38). In reality, the signal transients vary considerably for different applications and also for different data sequences of the same application. Therefore, numerous measured signals will be shown in Chapter 6 for the applications investigated. The diagnosis methods will be able to handle all these different signals. The input temperatures  $T_7$  and  $T_8$  at the heat source and at the heat sink side are also measured. They are regarded as input signals  $u_i$ . For the transients of the signals, the current operating mode is important (here ‘on’), as well as the operating mode immediately preceding (here ‘off’). Therefore the sequence is specified by both the current and the preceding operating mode (here ‘off-on sequence’).

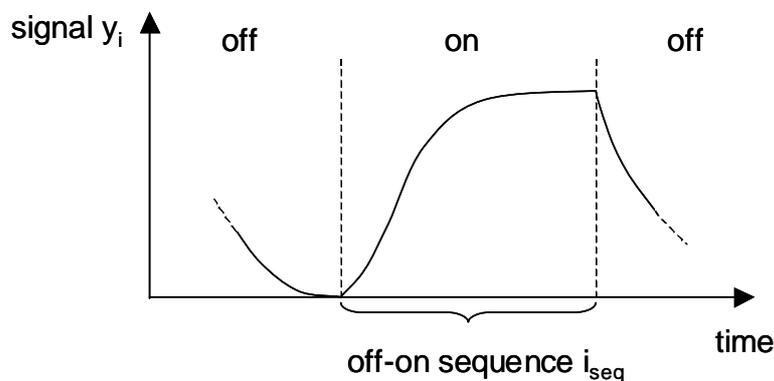


Fig. 10: Step response for one signal during the off-on sequence  $i_{seq}$

All measured output signals  $y_i(i_{seq})$  and input signals  $u_i(i_{seq})$  of one sequence  $i_{seq}$  are used to identify the parameters of a gray-box model structure (cf. Sections 5.2 and 5.3). Therefore the model structure is separated into submodels

$i$ , and the parameters  $\theta_i(i_{seq})$  of the submodels  $i$  are identified separately. The parameters  $\theta_i(i_{seq})$  of the submodels  $i$  are then combined into the parameter set  $\theta(i_{seq})$ .

The parameter set  $\theta(i_{seq})$  of a sequence  $i_{seq}$  defines one data point  $x(i_{seq}) = \theta(i_{seq})$  in the parameter space (cf. Fig. 11). The dimension of the parameter space is equal to the length of the parameter vector  $\theta(i_{seq})$ . The parameter identification step is repeated for each data sequence belonging to different fault cases. Thus, for each fault case, a multitude of data points is generated.

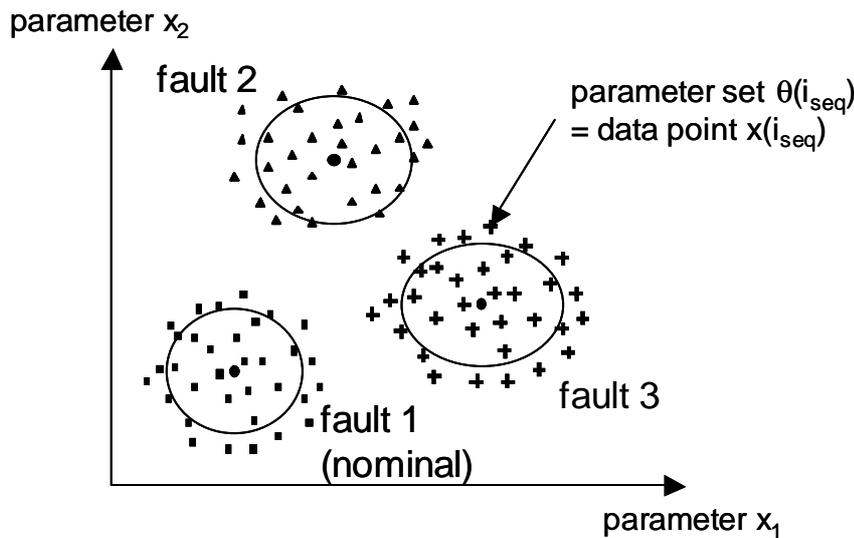


Fig. 11: Parameter space with fault clusters and data points. One data point corresponds to one parameter set.

In the next step, all data points belonging to one fault case are combined in one cluster. Generally about ten data points (from ten data sequences) are required for each fault case. The locations and the extensions of the clusters are calculated therefrom. With these fault clusters, a classification system is built. A typical time duration of the training is described in Section 12.3.1.

### 5.1.2 Operation

During operation, data sequences similar to the one shown in Fig. 10 are measured. With the same identification methods described above, the parameter sets  $\theta(i_{seq})$  are re-identified for each data sequence  $i_{seq}$ . Thus, one data sequence is represented by one data point in the parameter space. Afterwards, this data point is classified by the trained fault clusters, which means that the membership grades  $f_i(i_{seq})$  to all clusters  $i$  are calculated. The membership grades express the probabilities that the corresponding faults  $i$  are present. These probabilities are directly visualized by the software tool presented in Section 7.2.

## 5.2 Model

For modeling the refrigerant cycle, the structure of the model has to be chosen. For the general heat pump shown in Fig. 3 on page 38, the model has the input vector  $u(t)$  and output vector  $y(t)$  of eq. (1), containing the measured temperature signals  $T_i(t)$ . The additional step input signal  $u_{step}(t)$  is required because the heat pump is switched on and off. The output signals  $y(t)$  of the model are intensively excited by this step input.

$$y(t) = \begin{bmatrix} T_1(t) \\ T_2(t) \\ \vdots \\ T_6(t) \end{bmatrix} \quad u(t) = \begin{bmatrix} T_7(t) \\ T_8(t) \\ u_{step}(t) \end{bmatrix} \quad (1)$$

The measured input signals  $T_7(t)$ ,  $T_8(t)$ ,  $u_{step}(t)$  as well as the measured output signals  $T_1(t)$  and  $T_3(t)$  are plotted in Fig. 12 for a sample data sequence of the application II described in Section 6.1. This off-on data sequence is extracted from a measured data set containing several sequences. It starts at the time  $t = 1$  minute and ends at the time  $t = 14$  minutes.

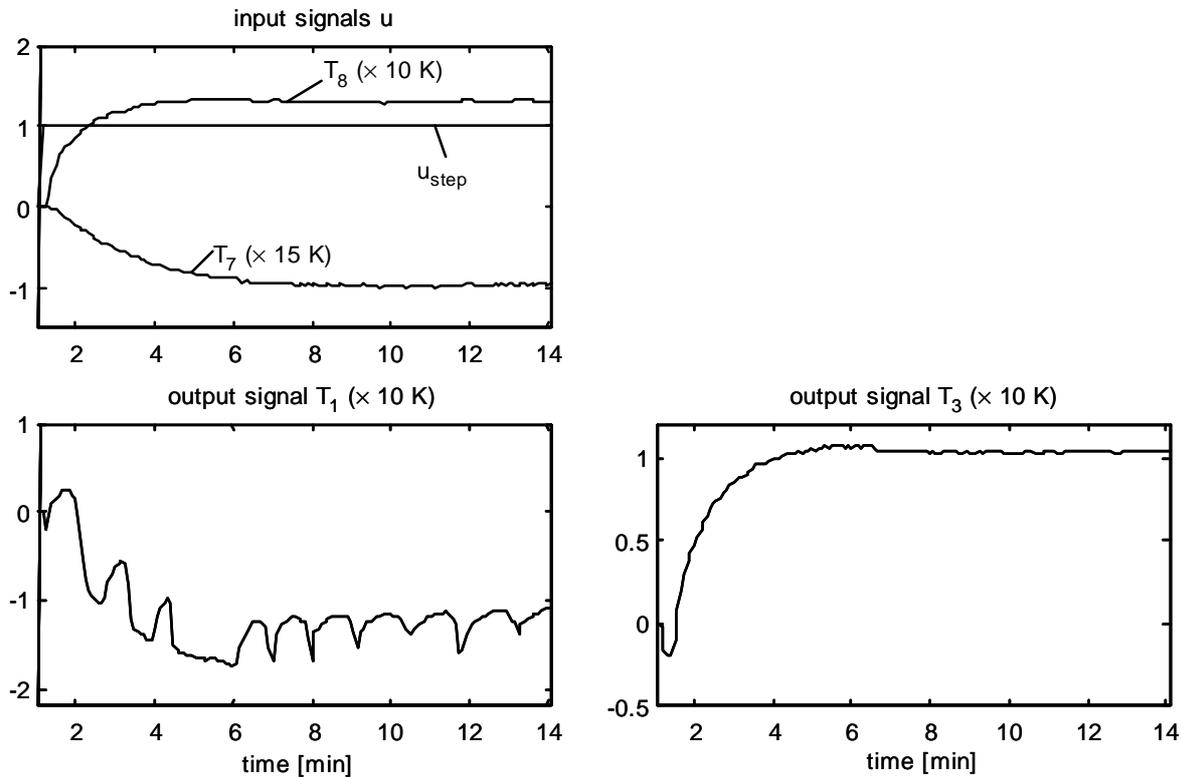


Fig. 12: Sample data sequence. Upper left plot: measured input signals ( $T_7$ ,  $T_8$ ) and step input signal  $u_{step}$ . Other plots: measured output signals  $T_1$ ,  $T_3$ . All signals are scaled: the (variable) offsets are set such that the signal values start at zero and the (fixed) gains are set to the values in the brackets ( $\times 10$  K,  $\times 15$  K).

Regarding one data sequence, the signal transients may be quite different for each output signal, which is shown here for two output signals. The output signal  $T_3$  is characterized by a fast overshoot in the negative direction at the beginning of the sequence and a slow transient afterwards. The output signal  $T_1$  is characterized by oscillations, which are superposed to the transients. The reason for those oscillations is the ‘hunting effect’ of the thermostatic expansion valve (cf. Section 4.1), which is especially strong in this application.

For one output signal (for example  $T_3$ ), the transients also change from one data sequence to another. These changes may contain information about the faults or they may be due to disturbances. Thus the task of fault diagnosis will be to extract information from the signal changes due to the faults and to ignore the variations due to disturbances.

### 5.2.1 Physical Structure (Gray-Box Model)

For the heat pump scheme shown in Fig. 3 on page 38, the model structure is given by the matrix (2) from physical considerations. Therefore, the ‘physical couplings’ between the signals are considered with respect to each module of the heat pump. Regarding the evaporator, its output signal  $T_1$  is coupled with the signals  $T_4$ ,  $T_5$  and  $T_7$ . These couplings are represented in the first row of the matrix, where the output signal  $y_1=T_1$  is coupled with the (output) signals  $T_4$ ,  $T_5$  and the input signal  $T_7$ , as well as with the additional input signal  $u_{step}$  (cells with ‘•’). Each row of the matrix describes the couplings of one output signal  $y_i$  with respect to one module.

$$\begin{array}{c|cccccc|cc}
 & T_1 & T_2 & T_3 & T_4 & T_5 & T_6 & T_7 & T_8 & u_{step} \\
 \hline
 T_1 & \times & 0 & 0 & \bullet & \bullet & 0 & \bullet & 0 & \bullet \\
 T_2 & \bullet & \times & 0 & 0 & 0 & 0 & 0 & 0 & \bullet \\
 T_3 & 0 & \bullet & \times & 0 & 0 & \bullet & 0 & \bullet & \bullet \\
 T_4 & 0 & 0 & \bullet & \times & 0 & 0 & 0 & 0 & \bullet \\
 T_5 & \bullet & 0 & 0 & \bullet & \times & 0 & \bullet & 0 & \bullet \\
 T_6 & 0 & \bullet & \bullet & 0 & 0 & \times & 0 & \bullet & \bullet
 \end{array} \tag{2}$$

The advantage of this model structure is its coherence to physics.

### 5.2.2 Decoupled Structure (Black-Box Model)

Without using any physical knowledge about the system, a decoupled structure is defined. Here the output signals  $T_1..T_6$  are not coupled among each

other (matrix (3)). On the other hand, each output signal is coupled with all input signals  $T_7$ ,  $T_8$ , and  $u_{step}$  (cells with ‘•’).

$$\begin{array}{c|cccccc|ccc}
 & T_1 & T_2 & T_3 & T_4 & T_5 & T_6 & T_7 & T_8 & u_{step} \\
 \hline
 T_1 & \times & 0 & 0 & 0 & 0 & 0 & \bullet & \bullet & \bullet \\
 T_2 & 0 & \times & 0 & 0 & 0 & 0 & \bullet & \bullet & \bullet \\
 T_3 & 0 & 0 & \times & 0 & 0 & 0 & \bullet & \bullet & \bullet \\
 T_4 & 0 & 0 & 0 & \times & 0 & 0 & \bullet & \bullet & \bullet \\
 T_5 & 0 & 0 & 0 & 0 & \times & 0 & \bullet & \bullet & \bullet \\
 T_6 & 0 & 0 & 0 & 0 & 0 & \times & \bullet & \bullet & \bullet
 \end{array} \tag{3}$$

With this model structure, the relation to physics is almost lost. But its advantage is that one or more output signals  $T_1..T_6$  can be easily eliminated by eliminating the corresponding rows and columns of the matrix. This feature will be important in order to reduce the number of sensors (cf. Section 5.5.1)

### 5.2.3 Full Structure (Black-Box Model)

Whereas the decoupled model structure is a simplification of the physical model structure, the full model structure is an extension of it. As mentioned above, the physical model structure is defined by considering the scheme on Fig. 3. But in the strict sense, this scheme is a simplification of the reality. In a real heat pump, the relative location of the modules is important. In a small brine-to-water heat pump for residential buildings, the plate heat exchangers of the evaporator and the condenser are located side by side, and all other modules as well as the connecting pipes are located in a narrow space. The sensors are also plugged to the pipes, typically wherever there is any space left. With the real configuration in mind, it is obvious that the sensors do not only measure the temperatures of the fluid at the locations shown in the scheme of Fig. 3, but they also measure the temperatures of the pipes or they may even be influenced by neighbouring pipes or modules. Thus, in reality, more couplings are given between the signals, even if they are smaller than the couplings of the ‘physical’ model structure. These additional couplings can be explained also for the case when the heat pump is shut down. Then the sensors in the upper part of the heat pump measure higher temperatures than the sensors in the lower part. In theory, all these additional couplings could be modelled. But without measuring their real magnitude, no decision can be made as to which couplings have to be added and which couplings are not important.

Therefore, a full model structure is defined in which all signals are coupled. Afterwards, the parameter identification algorithm of Section 5.3 will use

measured data to calculate, which couplings are important (large parameter values) and which are not important (low parameter values). In the matrix (4), the output signals  $T_1..T_6$  are fully coupled with all other output signals and all input signals  $T_7, T_8, u_{step}$ .

	$T_1$	$T_2$	$T_3$	$T_4$	$T_5$	$T_6$	$T_7$	$T_8$	$u_{step}$
$T_1$	×	•	•	•	•	•	•	•	•
$T_2$	•	×	•	•	•	•	•	•	•
$T_3$	•	•	×	•	•	•	•	•	•
$T_4$	•	•	•	×	•	•	•	•	•
$T_5$	•	•	•	•	×	•	•	•	•
$T_6$	•	•	•	•	•	×	•	•	•

(4)

The advantage of the full structure is its high fitting accuracy, because for each output signal, all available information of the other signals is used. But on the other hand, the danger of over-parametrization increases. This model structure is used in application IV described in Section 6.3, where only certain substructures for a low number of output signals are extracted. In that case, the number of parameters is kept low and there is no over-parametrization.

### 5.2.4 Submodels

The model structures of the matrices (2)...(4) are separated into submodels, which represent each row of the matrices with the corresponding output signals. For example, submodel 1 with the output signal  $T_1$  is represented by the 1<sup>st</sup> row of the matrices (2) ... (4).

For each submodel, a polynomial approach is selected for modeling the input-output behaviour. ARX models are well suited and can be identified in a straightforward manner (cf. Section 5.3). The discrete time representation of an ARX model is given by eq. (5) (AR = auto regressive, X = extra input) [Ljung 87], [Ljung 98]. The multiple-input single-output (MISO) system has the input vector  $u_i^*$  and the scalar output signal  $y_i$ . For an output signal  $y_i$  of the submodel  $i$  (row  $i$  of the system matrix), the vector  $u_i^*$  contains all coupled output signals  $y_j$  and input signals  $u_j$  (columns  $j$  of the system matrix with ‘•’ in the corresponding row  $i$ ).

$$\begin{aligned}
 y_i(t) + a_1 y_i(t-1) + a_2 y_i(t-2) + \dots + a_{n_a} y_i(t-n_a) = \\
 B_1 u_i^*(t-n_k) + B_2 u_i^*(t-n_k-1) + \dots + B_{n_b} u_i^*(t-n_k-n_b+1) + e_i(t)
 \end{aligned}
 \tag{5}$$

The parameters are represented by the scalars  $a_1 \dots a_{na}$  and the row vectors  $B_1 \dots B_{nb}$ , whereas the polynomial order is given by  $[n_a, n_b]$ , the number of time delays is given by  $n_k$ , and  $e_i(t)$  defines white noise.

Regarding a first order ARX model without time delay, eq. (5) is reduced to eq. (6) with  $n_a=1$ ,  $n_b=1$ ,  $n_k=0$ ,  $a=-a_1$ ,  $B=B_1$ .

$$y_i(t) = ay_i(t-1) + Bu_i^*(t) + e_i(t) \quad (6)$$

For example, an ARX model is defined by eq. (7) for submodel 1 of the physical structure, which corresponds to the 1<sup>st</sup> row of the matrix (2). The output signal  $y_1(t)=T_1(t)$  at time  $t$  is calculated from the signals  $u_1^*(t)=[T_4(t), T_5(t), T_7(t), u_{step}(t)]$  at time  $t$  (cells with ‘•’ in matrix (2)) and the past output  $y_1(t-1)=T_1(t-1)$  at time  $t-1$  (cell with ‘x’ in matrix (2)). This submodel contains the input and output signals of the evaporator and therefore represents the evaporator (cf. Fig. 3).

$$T_1(t) = a_1 \cdot T_1(t-1) + \begin{bmatrix} b_{1,4} & b_{1,5} & b_{1,7} & b_{1,step} \end{bmatrix} \cdot \begin{bmatrix} T_4(t) \\ T_5(t) \\ T_7(t) \\ u_{step}(t) \end{bmatrix} + e_1(t) \quad (7)$$

The evaporator is also represented by the submodel 5 corresponding to the 5<sup>th</sup> row of the matrix (2), with the output signal  $y_5(t)=T_5(t)$  and the coupled signals  $u_5^*(t)=[T_1(t), T_4(t), T_7(t), u_{step}(t)]$ . Here, the submodel output is defined by the heat source output temperature  $T_5(t)$  instead of the refrigerant output temperature  $T_1(t)$ , but for both submodels the same temperature signals  $T_1(t)$ ,  $T_4(t)$ ,  $T_5(t)$ ,  $T_7(t)$  at the interfaces of the condenser are used.

As another example, submodel 3 represents the condenser, with the refrigerant output temperature  $T_3(t)$  as the output signal (eq. (8), 3<sup>rd</sup> row of the matrix (2)).

$$T_3(t) = a_3 \cdot T_3(t-1) + \begin{bmatrix} b_{3,2} & b_{3,6} & b_{3,8} & b_{3,step} \end{bmatrix} \cdot \begin{bmatrix} T_2(t) \\ T_6(t) \\ T_8(t) \\ u_{step}(t) \end{bmatrix} + e_3(t) \quad (8)$$

The condenser is also represented by the submodel 6 corresponding to the 6<sup>th</sup> row of the matrix (2), with the output signal  $T_6(t)$  and the coupled signals  $[T_2(t), T_3(t), T_8(t), u_{step}(t)]$ .

Static models are used as well, where eq. (5) is simplified to eq. (9), with  $n_a=0$ ,  $n_b=1$ ,  $n_k=0$ ,  $K=B_I$ . Often, static approaches are sufficient for describing fast components of the process. Their advantage is that they have the least number of parameters and therefore can be identified under most any conditions.

$$y_i(t) = Ku_i^*(t) + e_i(t) \quad (9)$$

For the submodel 1 of the physical structure, the static model is defined by eq. (10).

$$T_1(t) = \begin{bmatrix} k_{1,4} & k_{1,5} & k_{1,7} & k_{1,step} \end{bmatrix} \cdot \begin{bmatrix} T_4(t) \\ T_5(t) \\ T_7(t) \\ u_{step}(t) \end{bmatrix} + e_1(t) \quad (10)$$

For the submodel 3 of the physical structure, the static model is defined by eq. (11).

$$T_3(t) = \begin{bmatrix} k_{3,2} & k_{3,6} & k_{3,8} & k_{3,step} \end{bmatrix} \cdot \begin{bmatrix} T_2(t) \\ T_6(t) \\ T_8(t) \\ u_{step}(t) \end{bmatrix} + e_3(t) \quad (11)$$

### 5.3 Parameter Identification

The parameters of the submodels are identified from the measured signals by using the parameter identification algorithm presented in Section 5.3.1. The algorithm is applied separately to each submodel (i.e., to each row of the matrices (2)...(4)) and to several model approaches (such as the static model approach and the first order ARX approach). The parameters are then used to simulate the submodels and to compare their output signals to the measured signals (Section 5.3.2). In Section 5.3.3, the best model approaches will be chosen for each submodel and in Section 5.3.4, the corresponding parameters will be analyzed regarding their sensitivities to the faults and their variations within each fault case.

### 5.3.1 Least-Squares Estimation

The ARX model permits the parameter identification task to be performed very simply, using the Least-Squares estimation method ([Ljung 87], [Söderström 89]). The linear regression form of an ARX model  $i$  is given by eq. (12), with the output  $y_i(t)$ , the regression vector  $\varphi_i^T(t)$ , the parameter vector  $\theta_i$ , and the white noise term  $e_i(t)$ .

$$y_i(t) = \varphi_i^T(t) \cdot \theta_i + e_i(t) \quad (12)$$

Now, the output  $y_i(t)$  and the regression vector  $\varphi_i^T(t)$  is built by measured data, which results in eq. (13) with the equation error  $\varepsilon_i(t)$ .

$$y_i(t) = \varphi_i^T(t) \cdot \theta_i + \varepsilon_i(t) \quad (13)$$

For example, the evaporator submodel  $i=1$  (eq. (7)) is transformed to eq. (14).

$$T_1(t) = \begin{bmatrix} T_1(t-1) & T_4(t) & T_5(t) & T_7(t) & u_{step}(t) \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ b_{1,4} \\ b_{1,5} \\ b_{1,7} \\ b_{1,step} \end{bmatrix} + \varepsilon_1(t) \quad (14)$$

Thus, the output  $y_1(t)$  and the regression vector  $\varphi_1^T(t)$  are defined by eq. (15) and the parameter vector is defined by eq. (16).

$$y_1(t) = T_1(t) \quad \varphi_1^T = \begin{bmatrix} T_1(t-1) & T_4(t) & T_5(t) & T_7(t) & u_{step}(t) \end{bmatrix} \quad (15)$$

$$\theta_1 = \begin{bmatrix} a_1 \\ b_{1,4} \\ b_{1,5} \\ b_{1,7} \\ b_{1,step} \end{bmatrix} \quad (16)$$

For  $n_k$  measurements, the linear regression form (eq. (13)) becomes eq. (17).

$$Y_i = \Phi_i^T \cdot \theta_i + E_i \quad (17)$$

For the evaporator submodel, the output vector  $Y_1$  and the regression matrix  $\Phi_1^T$  are defined by eqs. (18) and (19). An estimation for the parameter vector  $\theta_1$  is calculated by building the pseudo-inverse of  $\Phi_1^T$  (eq. (20)).

$$Y_1 = \begin{bmatrix} T_1(t) \\ T_1(t-1) \\ \vdots \\ T_1(t-n_k+1) \end{bmatrix} \quad (18)$$

$$\Phi_1^T = \begin{bmatrix} T_1(t-1) & T_4(t) & T_5(t) & T_7(t) & u_{step}(t) \\ T_1(t-2) & T_4(t-1) & T_5(t-1) & T_7(t-1) & u_{step}(t-1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ T_1(t-n_k) & T_4(t-n_k+1) & T_5(t-n_k+1) & T_7(t-n_k+1) & u_{step}(t-n_k+1) \end{bmatrix} \quad (19)$$

$$\hat{\theta}_{1,LS} = (\Phi_1 \Phi_1^T)^{-1} \Phi_1 \cdot Y_1 \quad (20)$$

For each type of a data sequence, another parameter set is identified. A general heat pump with two operating modes (off, on) has two types of data sequences (off-on, on-off). For an air-to-water heat pump with a defrosting cycle, three modes (off, on, defrosting) and six types of data sequences (off-on, off-defrosting, on-off, on-defrosting, defrosting-off, defrosting-on) have to be defined. Often it is sufficient to consider a subset of the sequence types or one sequence type only (cf. applications in Chapter 6). The off-on or defrosting-on sequences contain most information about the condition of the heat pump.

The persistency of the excitation and the consistency properties of the least-squares identification method are discussed in Appendix A.

### 5.3.2 Signal Plots for Submodels

The parameters of each submodel are identified from the measured signals. From the input signals  $T_7$ ,  $T_8$ ,  $u_{step}$  and the measured output signals  $T_1$ ,  $T_3$  of Fig. 12, the parameters of the submodels 1 and 3 are identified for the static model approaches and the first order ARX approaches by using the methods of Section 5.3.1. The submodels are then simulated using both approaches with the identified parameters. The simulated output signals are compared to the measured signals  $T_1$  and  $T_3$ . This procedure is repeated for each other submodel, such as the submodels 2, 4, 5 and 6.

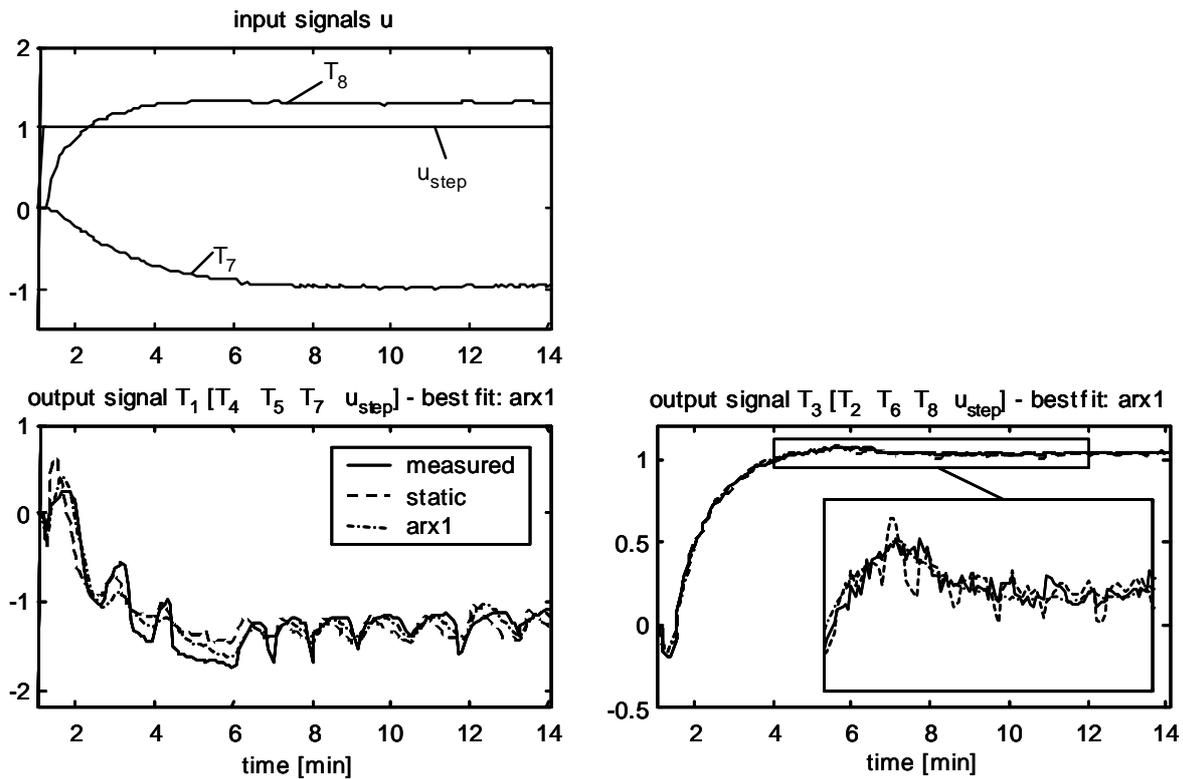


Fig. 13: Example data sequence, physical model structure. Upper left plot: measured input signals ( $T_7$ ,  $T_8$ ) and step input signal  $u_{step}$ . Other plots: measured (solid line) and simulated output signals  $T_1$ ,  $T_3$  for the static and first-order ARX approaches (arx1). All signals are scaled. The coupled signals are listed in brackets [], for example:  $T_1$  is coupled with the signals  $T_4$ ,  $T_5$ ,  $T_7$  and  $u_{step}$ . The approaches which best fit the measured data of the current sequence are noted in the title of the plots.

For submodel 3 with the output signal  $T_3$ , the static model approach as well as the ARX approach are close to the measured data, which means that the model errors are small. The ARX approach is even slightly closer (which can be seen in the zoomed box), but the difference between both approaches is very small. The ARX approach is smoother than the static approach because of its lowpass filtering effect. For the submodel 1 with the output signal  $T_1$  both approaches result in larger model errors. Here again, the ARX approach is slightly better, but with either approach it is more difficult to fit this signal.

The two output signals above are selected for showing a case of low model errors (output signal  $T_3$ ) and a case of high model errors (output signal  $T_1$ ). In the applications investigated, most signal fits are similar to that of the output signal  $T_3$  (cf. Chapter 6). Finding the ‘good’ output signals with low model errors is one task of the training procedure. The output signals  $T_1$  and  $T_3$  are not able to represent the multitude of different signal transients for all kind of submodels, fault cases and applications. Therefore, more signal plots are

shown in the Sections 6.1.3, 6.1.6, 6.2.3 and 6.3.3 for the corresponding applications. Nevertheless, some general discussion is made below.

From physical considerations, it would be expected that generally the dynamic ARX approach would fit significantly better than the static model approach because of the time constants in the thermal process. But for the sample output signal  $T_3$  this is not true, since the difference between the two approaches is very small. The reason for this is the fact that the time constants are already included in the transients of the coupled signals, which are measured. For example, the output signal  $T_3$  is coupled to the signals  $T_2$ ,  $T_6$  as well as to the input signal  $T_8$ . If some of the coupled signals have time constants that are in the same range, then the output signal  $T_3$  can be expressed by a linear combination of these signals. In other words, some of the coupling parameters  $k_{3,j}$  of the corresponding submodel will be large, namely those to the input signals  $j$  with similar transients, whereas the other parameters will be small. In the case of submodel 3, at least one signal has a transient similar to that of signal  $T_3$ , which is the input signal  $T_8$  in the upper left plot. Finding the values of the particular parameters will be the task of the identification algorithm in Section 5.3.1.

For the submodel 3, the step input signal  $u_{step}$  will only be weighted weakly by a small parameter value  $b_{3,step}$ . But for other output signals or other data sequences, the step input  $u_{step}$  might be very important, for example if the time constants of all coupled signals are much larger than the time constant of the output signal. Then, the step input is used for an extra excitation of the high frequencies. This is especially interesting in combination with the ARX approach in order to identify the time constants.

The output signal  $T_3$  is characterized by a fast overshoot in the negative direction at the beginning of the sequence and a slow transient until the time  $t = 9$  minutes. Afterwards, the signal is in a (quasi) steady state. The parameters of the submodel are influenced by the transient part of the signal, but also by the (quasi) steady-state part within the data sequence. The longer the relative length of the transient part is, the more the parameters are influenced by this part. By decreasing the total length of the data sequence, the relative length of the transient part is increased. This makes sense, if the important information (about the faults) is rather represented in the transient part than in the steady-state part. If the important information is represented in the steady-state part, then the total length of the data sequence has to be increased. Generally the disturbances also influence both parts, but in the applications investigated the transient part was more sensitive to them. Thus the transient part should not be weighted too much. Choosing an appropriate duration of a sequence is an

important design step. Here, a duration of 13 minutes is chosen ( $t = 1..14$  min.), which corresponds to a relative duration of the transient part at about 60% for the signal  $T_3$  (9-1 min. / 13 min.). For choosing the duration of a sequence, the signal with the largest time constant is taken into account.

Due to the ‘hunting effect’, the output signal  $T_1$  is characterized by oscillations, which are superposed to the slow transient as well as the quasi steady-state part of the signal. Since the oscillations in the coupled signals  $T_4$ ,  $T_5$ , and  $T_7$  are less strong than in the output signal  $T_1$ , the reconstruction of the signal  $T_1$  is difficult (for example, signal  $T_5$  is shown in Fig. 30, page 89). Nevertheless, the most important characteristics of the measured signal, such as the type and the frequency of the oscillations, are represented by the simulated signals.

The choice of the model structure also has an important influence on the model errors, since it defines the signals that are coupled to each output signal. Here, the physical structure is chosen, which considers the ‘physical couplings’ of the selected temperature signals, as shown in the scheme of Fig. 3 on page 38. By introducing additional couplings or using the full model structure, the output signal fits can be improved.

Regarding one specific output signal (for example  $T_3$ ), the signal shape within the transient part as well as the signal level within the steady-state part changes from one data sequence to another. These signal changes may contain information about the faults or they are due to disturbances. The variations due to disturbances are quite high for some particular signals (cf. signal plots in Section 6.1.3). Since the parameters of the submodels are identified from these signals, they also have rather high variations, which are due to varying signals. A systematic analysis of the signal sensitivities and variations (such as the signal-to-noise ratio) is replaced by the direct analysis of the identified parameters described in Section 5.3.4. The advantage of the direct analysis is that besides the signal variations due to the disturbances, it includes all other uncertainties due to modeling errors or parameter identification errors, as well.

In this Section, the model errors of one data sequence are analyzed. Since the measured signals vary from one data sequence to another, the model errors also vary. For a general error analysis, all available data sequences have to be taken into account, which will be done in the next Section.

### 5.3.3 Model Error Analysis

During the training of the fault diagnosis system, the best model approaches have to be chosen for each submodel. Therefore the model errors of all data sequences belonging to different fault cases are analyzed. For each data sequence  $i_{seq}$ , the model parameters are identified and the model errors are calculated by comparing the simulated output signals  $\hat{y}_{i_{seq},i,static}$  for the static approach and  $\hat{y}_{i_{seq},i,arx1}$  for the first-order ARX approach to the measured signals  $y_{i_{seq},i}$  (eqs. (21) and (22)). The model errors have already been discussed for the output signals  $y_{i_{seq},1=T_1}$  and  $y_{i_{seq},3=T_3}$  in Fig. 13.

$$Err_{i_{seq},i,static} = \frac{1}{n_k} \sum_{t=1}^{n_k} \left[ \hat{y}_{i_{seq},i,static}(\hat{\theta}_{i_{seq},i,static}, t) - y_{i_{seq},i}(t) \right]^2 \quad (21)$$

$$Err_{i_{seq},i,arx1} = \frac{1}{n_k} \sum_{t=1}^{n_k} \left[ \hat{y}_{i_{seq},i,arx1}(\hat{\theta}_{i_{seq},i,arx1}, t) - y_{i_{seq},i}(t) \right]^2 \quad (22)$$

For each submodel  $i$ , the mean quadratic errors  $Err_{i_{seq},i,static}$  and  $Err_{i_{seq},i,arx1}$  are calculated for the static approach as well as for the first-order ARX approach ( $arx1$ ). The variable  $n_k$  defines the number of samples, which is constant for all data sequences.

Since the data sequences belong to different fault cases, the results are separated by fault case. The errors  $Err_{i_{seq},i,static}$  and  $Err_{i_{seq},i,arx1}$  are visualized in the model error charts. Fig. 14 shows the resulting error charts for the application II of Section 6.1. Such error charts are calculated for all submodels  $i$ , but here only the charts for the submodels  $i=1$  and  $i=3$  are displayed. These plots show the statistics of the model errors over all data sequences, separated by fault case. In the present application, eight fault cases are investigated (including the nominal case = fault 1). For each fault case, one bar is plotted, which visualizes the model error statistics for the corresponding data sequences. One bar indicates the minimum and maximum errors (light gray), the mean error and the standard deviations of the errors (dark gray).

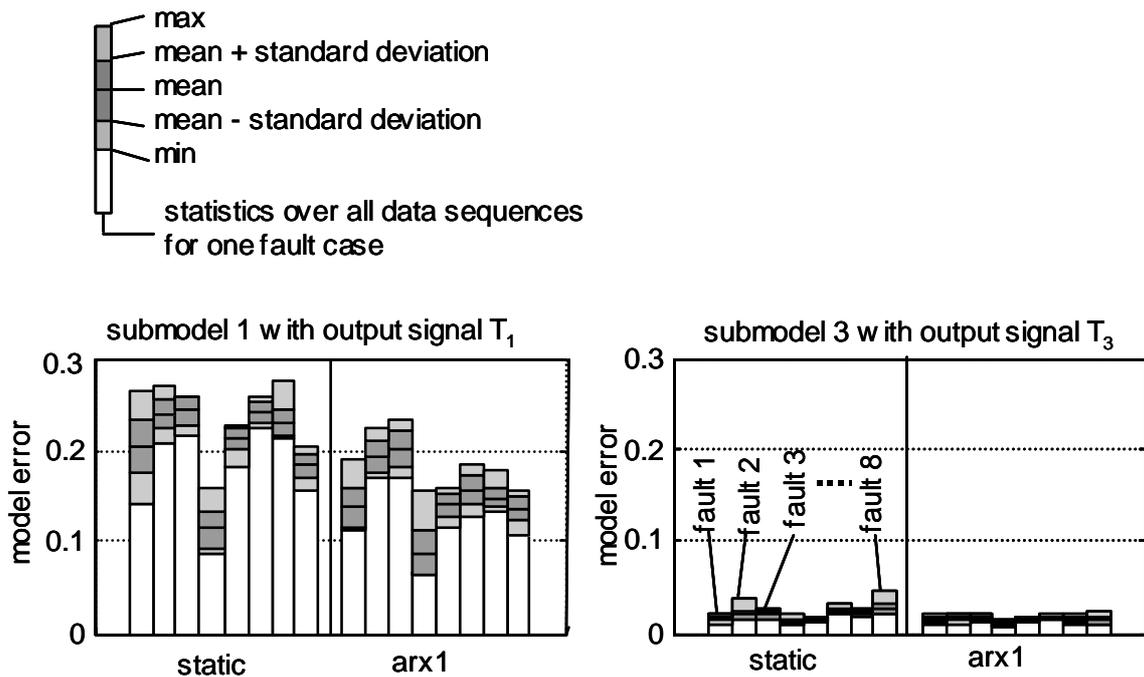


Fig. 14: Error charts for the submodels 1 and 3 with the output signals  $T_1$  and  $T_3$ , physical model structure. For each plot, the errors of the static model are in the bar group on the left, and the errors of the 1<sup>st</sup> order ARX model are in the bar group on the right; within a group, each bar represents one fault case (faults 1..8 from left to right). Legend for the bars (on the upper left): light gray region = minimum/maximum values of the errors, dark gray region = standard deviations of the errors

Regarding the chart for submodel 3, the left bar group represents the errors of the static model approach and the right bar group represents the errors of the first-order ARX model approach. Each bar group contains eight bars, representing the eight fault cases. In this case, both the ARX approach and the static model approach result in low model errors. Apparently, the observations made with the output signal  $T_3$  for one data sequence as shown in Fig. 13 are also valid for all other data sequences. The model errors vary little between the different fault cases.

Generally it is important to analyze all model errors for all fault cases during the training of the diagnosis system. If the maximum model error of only one fault case is too large, then the reliability of the diagnosis system is impaired. Having low model errors for all fault cases means that the output signals of the submodels fit all (trained) kinds of different signal transients for all fault cases. In other words, the submodels then are able to represent the input-output behaviour for all trained fault cases. The lower the model errors and the lower the variations of the model errors are within one fault case, the higher the robustness against disturbances will be.

Regarding the chart for submodel 1, both the ARX approach and the static model approach result in larger model errors, which has also been observed with the output signal  $T_I$  for one data sequence on Fig. 13. Here the model errors vary more between the different fault cases.

Now, for each submodel a decision has to be made as to which approach will be selected. For submodel 3 in Fig. 14, either the static approach or the ARX approach can be selected, since the errors are low for both approaches. In application II of Section 6.1, the ARX approach has been selected. Submodel 1 has larger errors for both approaches. Usually such a submodel would be eliminated, which means that it would not be used for fault classification. But in application II, the ARX approach (with the smaller errors) has been selected for that submodel, because the corresponding output signal  $T_I$  is able to represent the most important characteristics of the measured signal (cf. Section 5.3.2) and it contains important information about the faults.

For all other submodels, such as the submodels 2, 4, 5, and 6, the same analysis is made. As a result of the model error analysis, an appropriate approach is selected for each submodel and, if necessary, the number of submodels is reduced by elimination. A reduction of the number of submodels is not automatically related to a reduction of the ‘physical substance’ of the process description. Section 5.2.4 showed that some modules of the heat pump are represented by two submodels. Using the physical model structure, the evaporator is represented by the submodels 1 and 5, and the condenser is represented by the submodels 3 and 6. Thus, one submodel each can be eliminated without any loss of substance.

In the next Section, the parameters corresponding to the selected submodel approaches will be extracted and analyzed.

### 5.3.4 Sensitivity Analysis

In this Section, the sensitivities of the parameters to the faults and the variations of the parameters within each fault case are analyzed. Although the sensitivity analysis is not directly used for the training of the fault classification system described in the next Section, it is useful for estimating the quality of the diagnosis by the visualization in the parameter charts.

In theory, for each fault case  $f_i$  there exists one true parameter vector  $\theta_{0,i}$ . In the presence of unknown inputs (disturbances, noise) and model uncertainties, the estimated parameter vectors may vary according to eq. (23), where  $\Delta\theta_{0,i}$  denotes the estimation bias and  $\delta\theta_i$  denotes the uncertainty [Gertler 98].

The least-squares estimation method may result in biased parameters (cf. Appendix A).

$$\hat{\theta}_i = \theta_{0,i} + \Delta\theta_{0,i} + \delta\theta_i \quad (23)$$

For the fault diagnosis system presented here, the true values  $\theta_{0,i}$  of the parameters are not important. Therefore eq. (23) is simplified to eq. (24), where the bias is included in the mean parameter vector  $\bar{\theta}_i$  and in the uncertainty  $\delta\theta_i$ . The mean parameter vector  $\bar{\theta}_i$  is a function of the fault  $f_i$ . By linearizing the sensitivity function  $h(f_i)$ , the parameter change  $\Delta\bar{\theta}_i$  can be expressed by the size of the fault  $\Delta f_i$ .

$$\hat{\theta}_i = \bar{\theta}_i + \delta\theta_i \quad \bar{\theta}_i = h(f_i) \quad \Delta\bar{\theta}_i \approx \frac{\partial h}{\partial f_i} \Delta f_i \quad (24)$$

For an optimal diagnosis quality, the uncertainties  $\delta\theta_i$  should be minimal and the sensitivities  $\partial h/\partial f_i$  to the different faults  $i$  should be at a maximum.

For the application II of Section 6.1, the uncertainties and the sensitivities are shown in the parameter charts of Fig. 15. Such charts are plotted for the selected approaches for each submodel. In Section 5.3.3, the ARX approaches have been selected for the submodels 1 and 3, and therefore, the parameters of the ARX approaches are plotted on Fig. 15. Similar plots will be made for the submodels 2, 4, 5, and 6.

For each of the nine fault cases, one bar is plotted, which indicates the minimum and maximum parameters (light gray region), the standard deviations of the parameters (dark gray region), and the mean values of the parameters (in the middle of the dark gray regions). All parameters are scaled to the interval 0..1. For the submodel 1, the ARX approach contains five parameters (eq. (7)), which are represented by one bar group each. For the output signal  $T_3$ , the corresponding ARX model also contains five parameters (eq. (8)). The overlappings between the bars should be small, which is equivalent to a high sensitivity and a low variance.

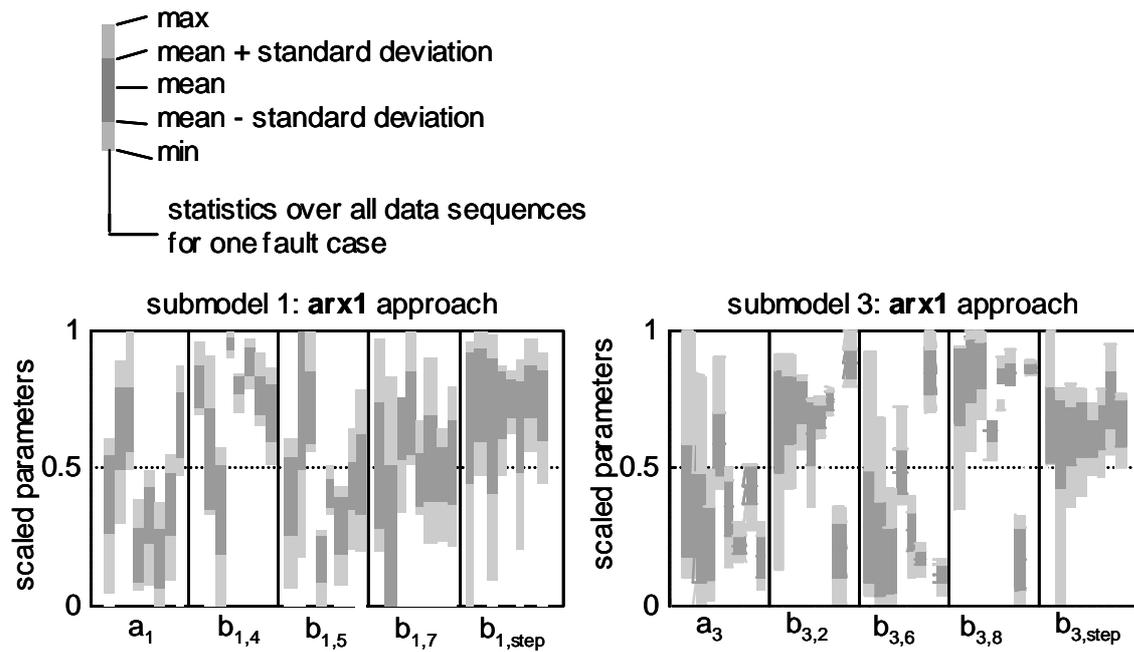


Fig. 15: Parameter charts for the submodels 1 and 3 with the first-order ARX approaches selected. Within a plot, each bar group represents one parameter (plot for submodel 1: parameters  $a_1$ ,  $b_{1,4}$ , ... from left to right); within a group, each bar represents one fault case (faults 1..8 from left to right). Legend for the bars (on the upper left): light gray region = minimum/maximum parameter values, dark gray region = standard deviations of the parameters. All parameters are scaled.

The variations of the parameters are mainly due to varying signal transients between the data sequences, also including varying oscillations, for example in the signal  $T_I$ , due to the ‘hunting effect’ of the thermostatic expansion valve (cf. signal plots in Section 6.1.3). Other effects such as additional noise, model uncertainties, or the bias of the parameter identification are much less severe. Even with these parameter variations, which are mainly given by the physics of the process, the faults can be classified with the methods described in the next Section.

By analyzing the bar groups, some ‘good’ parameters with high sensitivities to all or some of the fault cases can be found, such as the parameters  $a_1$ ,  $b_{1,4}$ ,  $b_{1,5}$ ,  $a_3$ ,  $b_{3,2}$ ,  $b_{3,6}$ ,  $b_{3,8}$ . But by only considering one of these parameters, a complete separation of all fault cases is not possible. Therefore, all of the parameters of the selected approaches for all submodels are combined in the next Section. Thus, the one-dimensional analysis for each single parameter will be extended to a high-dimensional analysis in the parameter space. For the two submodels in the example above, all ten parameters  $a_1$ ,  $b_{1,4}$ ,  $b_{1,5}$ ,  $b_{1,7}$ ,  $b_{1,step}$ ,  $a_3$ ,  $b_{3,2}$ ,  $b_{3,6}$ ,  $b_{3,8}$ ,  $b_{3,step}$  of both ARX approaches would be combined in a ten-dimensional parameter space. In reality, the parameters of the selected ap-

proaches for the submodels 2, 4, 5, and 6 are also combined in the same parameter space, which results in a higher dimensionality. Regarding the structure matrix (2) on page 48, the number of parameters can be counted. Supposing that for all submodels 1..6, the ARX approach was selected, then the number of parameters would be  $n = 26$  (counting all ‘•’ and ‘×’ in the structure matrix (2)) For the applications in Section 6, the dimension  $n$  of the parameter space is even higher, since the model structures contain more parameters.

## 5.4 Fault Classification Training

The one-dimensional analysis for each parameter presented in the Section above is now replaced by a clustering algorithm in a high-dimensional parameter space. Therefore, the  $n$ -dimensional parameter space is attempted to be separated into  $n_c$  clusters with one cluster for each fault case ( $n = 26$  and  $n_c = 8$  for the example above). Based on measured data, the classification system is trained in two steps.

In the first step the clusters are built. Therefore, several clustering methods have been tested. Sections 5.4.1 and 5.4.2 consider ‘hard’ clustering methods, where the data points (parameter sets) either have the weights 1 or 0. Section 5.4.3 considers ‘soft’ clustering methods, where the data points have continuous weights between 0 and 1.

Based on these clusters, a classification system is built in the second step. The resulting classification systems are ‘soft’ or ‘fuzzy’, which means that continuous membership grades between 0 and 1 are assigned to the data points (parameter sets) during operation. The membership grades specify the possibilities that the trained faults are present. For one data point, the number of membership grades is equal to the number of fault cases. Comparing to ‘hard’ classifications, these fuzzy systems are more robust and result in fewer wrong classifications.

### 5.4.1 HCM2Fuzzy Method

During the first step of the training, the clustering method Hard-C-Means (HCM) calculates the centers (arithmetic mean values) and standard deviations for each fault cluster on each parameter axis direction (Fig. 16).

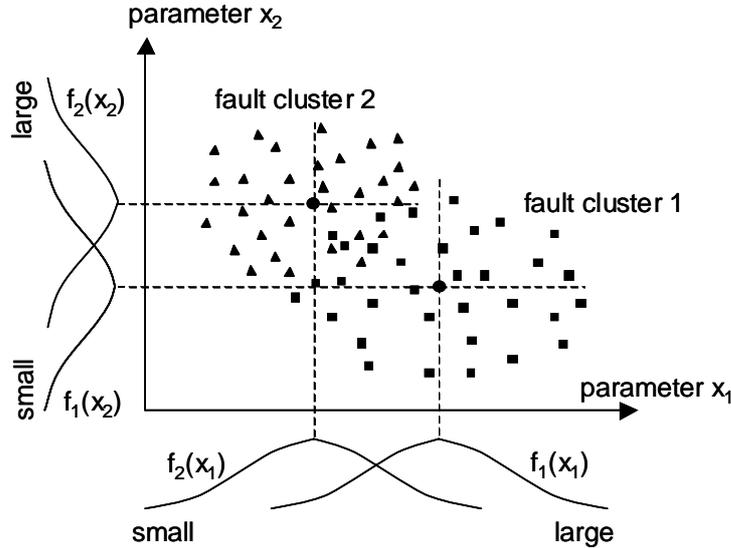


Fig. 16: Hard-C-means (HCM) clusters for two parameters and two faults with a projection onto four Gaussian fuzzy membership functions.

This clustering is called ‘hard’ because for the calculation of one cluster center, all data points corresponding to the corresponding fault case have the weight 1 and all other data points have the weight 0. In the second step, a fuzzy classification system is built by projecting the clusters onto the axis. A Gaussian membership function for the fault  $i$  is defined by the center  $c_{i,k}$  and the standard deviation  $\sigma_{i,k}$  (cf. eq. (25)),  $x_k$  being the component of a data point  $x$  in the direction of the axis  $k$ .

$$f_i(x_k) = e^{-\frac{(x_k - c_{i,k})^2}{2\sigma_{i,k}^2}} \quad (25)$$

For two fault clusters of Fig. 16 the extracted rules are:

*IF* ( $x_1$  is large) *AND* ( $x_2$  is small) *THEN* (*fault1* = yes) (*fault2* = no)

*IF* ( $x_1$  is small) *AND* ( $x_2$  is large) *THEN* (*fault1* = no) (*fault2* = yes)

The rules of the fuzzy inference system (FIS) can be viewed or edited by an expert. Building the system, appropriate methods for the FIS operators have to be chosen [Jang 99], [Geering 98].

During operation, a data point  $x$  is classified by calculating the membership grades  $f_i(x_k)$  for each component  $x_k$  to each fault cluster  $i$  using eq. (25) (fuzzification), and by applying the fuzzy AND operators to the resulting membership grades. The subsequent FIS operators are then applied, such as the implication, aggregation, and defuzzification operators [Jang 99].

For the application II of Section 6.1, Fig. 17 shows the training data points as well as the membership grades (shading) for one cluster of the resulting classification system. In that application, there are eight fault clusters in the parameter space; Fig. 17 depicts a projection of one cluster onto a plane of two dimensions. Since the training parameters are scaled to the range 0..1, the training data points will also be in this range. But the distribution of the membership grades is also interesting outside of this range, since during operation the data points might be outside as well. For this visualization, a range of  $-0.2..1.2$  is used. Here a rectangular cluster shape is recognizable, which is a result of the FIS operators selected. Here, the ‘min’ function has been chosen for the AND operator as well as the implication operator, the ‘max’ function has been chosen for the aggregation operator, and a ‘centroid’ defuzzification method has been used.

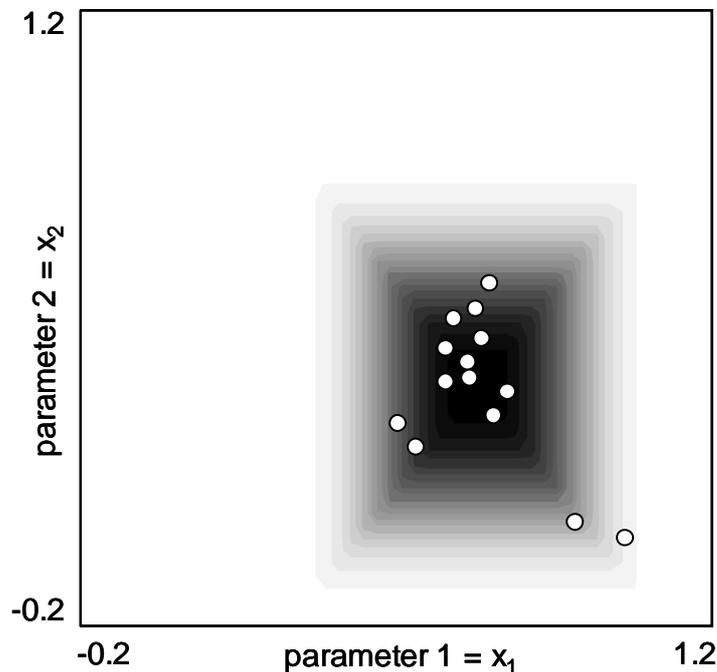


Fig. 17: Results of the HCM2Fuzzy method, cut through the center of cluster  $i=3$  (fault 3) of 8 clusters, projection from  $n$  dimensions onto the first two parameter axes. Dark = high membership grades, white = low membership grades. Data points = training parameter sets.

The advantage of the HCM2Fuzzy clusters is their high adaptability to the training data points (training parameter sets). The training data points are well represented by the shape of the cluster, which is rather ‘sharp’. But during operation, the locations of the data points (parameter sets) will differ from the locations of the training data points, mainly due to disturbances in the signals. Since the shape is rather ‘sharp’, these operational data points might have much lower membership grades, even if they are only a little displaced. Thus, the ‘robustness’ of these clusters is not very high. Here, the robustness is defined as the ability to classify the data points correctly, even if their locations vary due to disturbances.

Alternatively, a FIS system can be regarded as a neural network and inherits its learning capabilities (Fig. 18). Similar nets are the ANFIS (adaptive neuro-fuzzy inference systems [Jang 99]), where the input membership functions as well as the weightings of the rules can be adapted by learning algorithms.

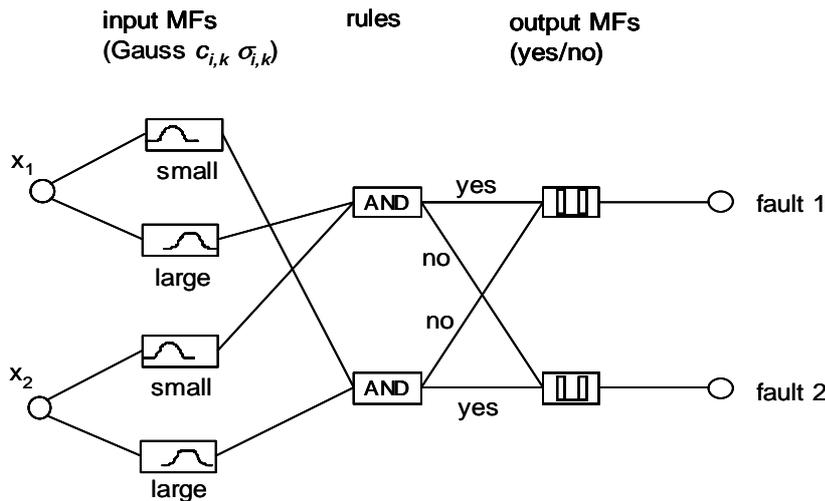


Fig. 18: A fuzzy inference system (FIS) in form of a neural network with two inputs, four Gaussian input membership functions, two rules, two output membership functions (yes/no), and two outputs.

### 5.4.2 HCM2Neuro Method

The HCM clusters can also be directly included in a neural network classification system. For each cluster  $i$ , the corresponding neuron transfer function is defined by eq. (26), with the cluster center components  $c_{i,1} \dots c_{i,n}$  and the standard deviation components  $\sigma_{i,1} \dots \sigma_{i,n}$ . Setting the exponential factor  $a$  to 2, the Gaussian function of eq. (25) is extended to  $n$  dimensions.

$$f_i(x_1 \dots x_n) = \prod_{k=1}^n e^{-\frac{(x_k - c_{i,k})^2}{a\sigma_{i,k}^2}} \quad (26)$$

For two parameters and two faults the resulting neural network is shown in Fig. 19. The advantage of these neural networks is their simple structure, which results in a fast evaluation.

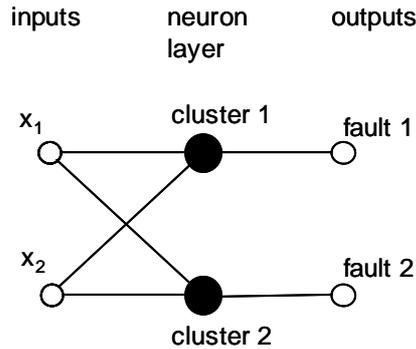


Fig. 19: A neural network with two inputs, two neurons for each cluster, and two outputs.

For the application II of Section 6.1, Fig. 20 shows the resulting ellipsoid shape of cluster 3 with different standard deviations in either component direction.

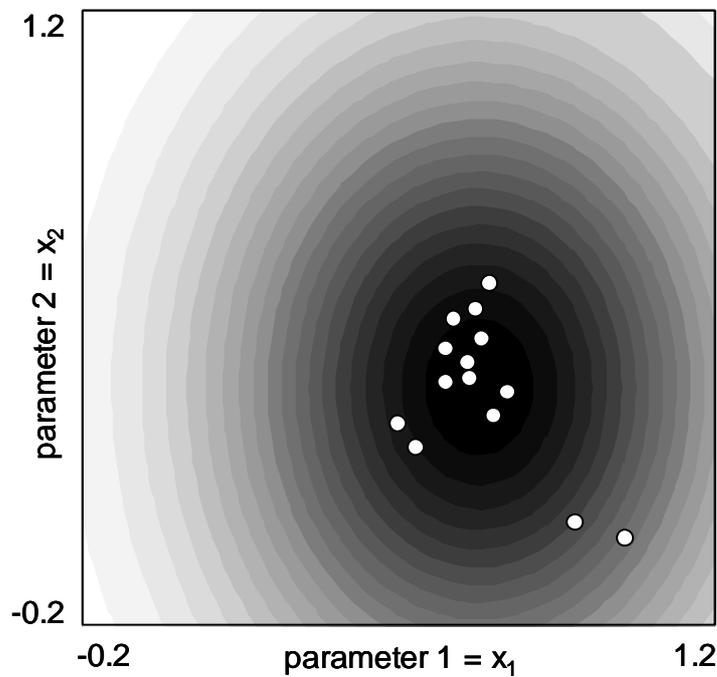


Fig. 20: Results of the HCM2Neuro method, cut through the center of cluster  $i=3$ . Data points = training parameter sets.

The training data points are also represented well by the HCM2Neuro cluster. Comparing Fig. 20 to Fig. 17, the higher ‘spread’ of the HCM2Neuro cluster is obvious. During operation, the robustness of this cluster to varying locations of the data points will be higher. Additionally, the spread can be influenced by the cluster parameter  $a$  (eq. (26)).

The net based on eq. (26) is similar to the radial basis function (RBF) net of eq. (27). The shape of an RBF cluster would be spherical, because the Euclidean distance between the data point  $x=[x_1..x_n]$  and the center  $c_i$  is used for the calculation of the membership grade.

$$f_i(x) = e^{-\left(\frac{\|x-c_i\|}{b}\right)^2} \quad (27)$$

RBF nets are often used for classification tasks ([Koivo 94], [Guglielmi 95]).

### 5.4.3 FCM2Neuro and PFCM2Neuro Methods

The idea of this method is to use the Fuzzy-C-Means (FCM) clustering algorithm to build a classification system ([Ayoubi 96], [Chiu 94]). This algorithm finds the optimal cluster centers by iteratively minimizing an objective function. The standard ‘unsupervised’ FCM algorithm ([Höppner 97], [Jang 99]) is designed to find clusters in a number of unclassified data points. The approach presented here is a modification of that approach. Since the classification of the data points (i.e., the assignment of the data points to the appropriate fault clusters) is already known from experiments, the modification to a ‘supervised’ algorithm is reasonable.

A data point  $x$  is classified by the membership grade of eq. (28). Regarding cluster  $i$ , the membership grade is a function of the distance  $d_i$  (eq. (29)) to the actual cluster center  $c_i$  and of the distances  $d_j$  to all other  $n_c$  cluster centers  $c_j$ . The parameter  $m$  influences the shape of the cluster.

$$f_i(x) = \frac{1}{\sum_{j=1}^{n_c} \left( \frac{d_i^2(x)}{d_j^2(x)} \right)^{\frac{1}{m-1}}} \quad (28)$$

$$d_i(x) = \|x - c_i\| \quad (29)$$

This means that the shape of an FCM cluster  $i$  is influenced by all other clusters  $j$ , as well.

The first step of the iterative FCM algorithm is to calculate the membership grades using eqs. (28) and (29) for all data points of each cluster  $i$  with some initial centers  $c_{i,0}$ . The objective function  $J$  of eq. (30) is evaluated therefrom. It measures the weighted quadratic mean distance between the points and the centers. In the next step, the new cluster centers  $c_i$  are determined by the weighted sum of eq. (31) over  $l=1..n_p$  points of each cluster.

$$J = \sum_{i=1}^{n_c} \sum_{l=1}^{n_p} f_i^m(x_l) \cdot d_i^2(x_l) \quad (30)$$

$$c_i = \frac{\sum_{l=1}^{n_p} f_i^m(x_l) \cdot x_l}{\sum_{l=1}^{n_p} f_i^m(x_l)} \quad (31)$$

The iteration (28) - (30) - (31) - (28) - ... is stopped when the improvement of  $J$  from one step to the next is smaller than a given value  $\varepsilon$ .

In the ‘supervised’ algorithm with eq. (31), the center  $c_i$  of cluster  $i$  is calculated only from the points  $l=1..n_p$  of the same cluster  $i$  rather than from all points. The initial centers  $c_i$  are calculated by the HCM clustering method (Secs. 5.4.1, 5.4.2), thus by the arithmetic mean value.

The resulting shape of the *FCM2Neuro* cluster in Fig. 21 (for the application II of Section 6.1) is influenced by the corresponding training data points as well as by the locations of the other clusters. The fewer other clusters exist next to the actual cluster, the more spherical its shape becomes [Höppner 97].

The advantage of the *FCM2Neuro* clusters is their higher ability to separate fault clusters that are close to each other. Since the shape of a cluster is influenced by the other clusters close to it, the overlappings between the clusters can be minimized. On the other hand, the centroid of the cluster might be displaced in comparison to the training data points, if it is pushed in a certain direction by the other clusters, as shown in Fig. 21. Therefore, the classification quality might be lower for the training data points (as well as the operational data points). But during operation, the clusters will be robust because of their large spread. With the cluster parameter  $m$  (eq. (28)), the spread and the shape of the cluster can be influenced.

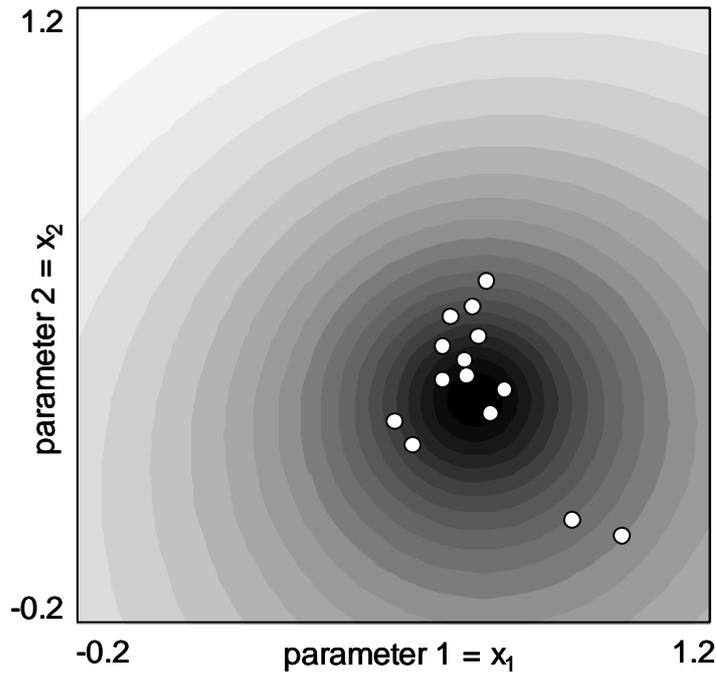


Fig. 21: Results of the FCM2Neuro method, cut through the center of cluster  $i=3$ . Data points = training parameter sets.

The probabilistic FCM definition of eq. (28) can be replaced by the possibilistic PFCM definition of eq. (32), which contains the parameter  $r_i$  for the extent of cluster  $i$  [Höppner 97]. Here the membership grades of cluster  $i$  are independent of the other clusters.

$$f_i(x) = \frac{1}{1 + \left( \frac{d_i^2(x)}{r_i^2} \right)^{\frac{1}{m-1}}} \quad (32)$$

A ‘supervised’ design algorithm similar to the one described in the case of the FCM method can be used (see above).

With the *PFCM2Neuro* approach, spherical cluster shapes are obtained, as shown in Fig. 22.

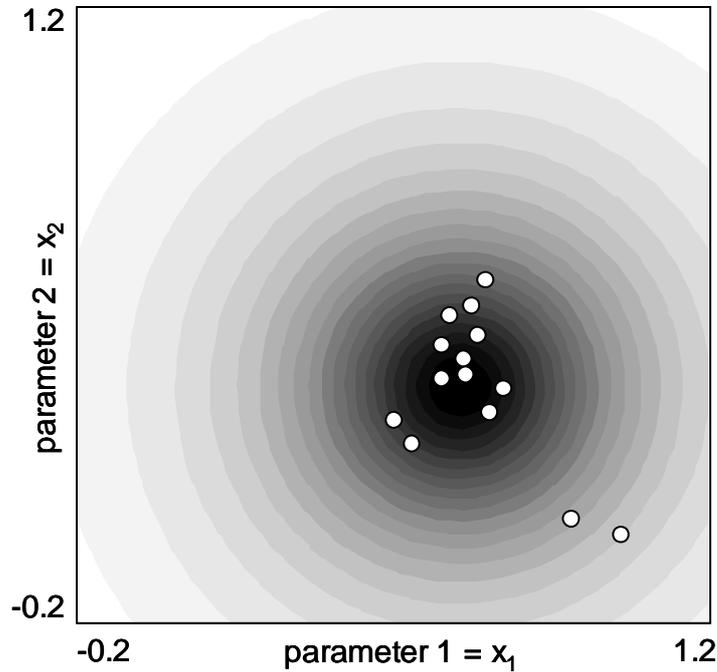


Fig. 22: Results of the *PFCM2Neuro* method, cut through the center of cluster  $i=3$ . Data points = training parameter sets.

There is no displacement of the cluster centroid, since the actual cluster is only influenced by the corresponding training data points. But the probability of cluster overlappings is higher than with FCM, since the clusters are created independently. Here the cluster parameters  $m$  and  $r_i$  (eq. (32)) can be used to influence the spread of the cluster, which is in analogy to the parameter  $a$  of the HCM2Neuro clusters (eq. (26)). Comparing this method to the suboptimal HCM2Neuro clustering method, the optimal cluster centers  $c_i$  here are found subject to the objective function  $J$ , which is minimized for the training data points (cf. FCM iteration above). Since the locations of the data points may vary during operation, this optimality is partially lost.

No general statement can be made as to which one of the clustering methods described above will be best suited for a particular application. With the software tool introduced in Section 7.1, all methods above can be applied and compared for any given training data set. As a result of the training procedure, the best methods will be selected. For all applications investigated in Section 6, the HCM2Neuro method results in a high classification quality, even for data sets that are not identical to the training data sets. Therefore, the HCM2Neuro method is recommended to be used in a first run.

#### 5.4.4 Cluster Checking

In order to check the classification quality of the trained clusters and to compare different cluster types, different model structures, or different submodel selections during the training phase, an appropriate measure has to be defined.

The *fault classification quality* is given by [Gertler 98] as

- the false alarm rate and
- the missed detection rate.

For a data point  $x_{nom}$  belonging to the nominal case, a false alarm occurs when the membership grade  $f_i$  to any fault cluster  $i$  exceeds the membership grade  $f_{nom}$  to the nominal cluster (eq. (33)).

$$f_i(x_{nom}) > f_{nom}(x_{nom}) \quad (33)$$

For a data point  $x_i$  belonging to any fault case  $i$ , a missed detection occurs when the membership grade  $f_{nom}$  to the nominal cluster exceeds the membership grade  $f_i$  to the fault cluster  $i$  (eq. (34)).

$$f_{nom}(x_i) > f_i(x_i) \quad (34)$$

Regarding two fault clusters  $i$  and  $j$ , a wrong classification rate can be defined analogously (eq. (35)). For this definition, the nominal cluster is referred to as one of the fault clusters (typically  $f_{nom}=f_1$ ). For a data point  $x_i$  belonging to any fault case  $i$ , a wrong classification occurs when the membership grade  $f_j$  to another fault cluster  $j \neq i$  exceeds the membership grade  $f_i$  to the fault cluster  $i$ .

$$f_j(x_i) > f_i(x_i) \quad (35)$$

This checking is repeated for each data point  $x_i$  of each cluster  $i=1..n_c$ . If eq. (35) is fulfilled, the number of wrong classifications  $n_{wrong}$  is increased by 1. At the end, eq. (36) yields the rate of wrong classifications, with  $n_{check}$  as the total number of checkings.

$$r_{wrong} = \frac{n_{wrong}}{n_{check}} \quad (36)$$

The rate of wrong classifications should be minimal. The results for the applications are given in Chapter 6. Note that this algorithm checks each single data point (parameter set) for a wrong classification. By averaging several data points (parameter sets) before classification, the rate of wrong classifications can be minimized significantly during operation (cf. Section 5.7.2). For

comparing the results during the training phase, no averaging is used. Thus the checking results of Sections 6.1 and 6.2 are “worst case” results, which define the upper bound of wrong classifications without averaging. In Section 6.3, the results for online operation with averaging are presented as well.

## 5.5 Minimizing the Number of Sensors

During the training of the fault diagnosis system the number of required sensors has to be minimized. For this purpose, the signals with the largest amount of ‘fault information’ have to be found.

Two different methods are suggested for this purpose. The method of Section 5.5.1 focuses on the submodels of the system matrices, especially of the physical structure in Section 5.2.1 or the decoupled structure in Section 5.2.2. The selected model structure is not modified by this method. In contrast, the method of Section 5.5.2 modifies the model structure in order to select a subset of signals. It can be applied to all model structures, but it is especially straightforward with the full structure of Section 5.2.3.

### 5.5.1 Checking Combinations of Submodels

This method finds the signals with the largest amount of ‘fault information’ by finding the corresponding submodels. For example, the physical structure matrix (2) on page 48 contains the six submodels  $i=1..6$ , corresponding to the six rows  $i=1..6$ . As a result of the error analysis described in Section 5.3.3, some submodels with high model errors might already have been eliminated (depending on the application investigated). But in order to simplify the further explanations, no submodels are assumed to be eliminated and all submodels 1..6 are assumed to be represented by the ARX approach.

For all data sequences, the parameter sets have already been identified for all submodels by using the methods of Section 5.3.1. They are stored in a parameter matrix, with each row containing one parameter set for one data sequence, and each column corresponding to one model parameter. For all submodels with ARX approaches, there are  $n=26$  columns.

Now, only a reduced number of submodels is selected, such as the combinations of two submodels (1, 2), (1, 3), (1, 4),..., (5, 6). For example, the combination (1, 3) of the submodels 1 and 3 is selected, which correspond to eqs. (7) and (8) on page 51.

Together, both submodels have a total number of ten model parameters (eq. (37)).

$$\theta_{1,3} = [a_1 \quad b_{1,4} \quad b_{1,5} \quad b_{1,7} \quad b_{1,step} \quad a_3 \quad b_{3,2} \quad b_{3,6} \quad b_{3,8} \quad b_{3,step}] \quad (37)$$

From the parameter matrix, the corresponding ten columns are extracted. The extracted matrix contains the reduced parameter sets for all data sequences. These parameter sets represent the data points in a parameter space, which is reduced to ten dimensions. Based thereon, a classification systems is built by using one of the clustering algorithms of Section 5.4. It is then checked for wrong classifications using the algorithm of Section 5.4.4. This procedure is repeated for all combinations of two submodels. At the end, the results of these checkings are compared, yielding the submodel combination with a minimal rate of wrong classifications. The algorithm described so far has been automated by a special software tool.

In other words, that submodel combination is selected which is best suited to separate all fault cases in the corresponding (reduced) parameter space. Since each submodel  $i$  corresponds to one output signal  $T_i$  and to a list of coupled signals (according to Section 5.2.4), the best signals are automatically selected with the selection of the best submodels. If the number of wrong classifications is too high for all combinations of two submodels, then the checking is repeated for combinations of three or more submodels. In special cases, the number of selected submodels might be reduced to one.

The efficiency of the sensor reduction algorithm depends on the model structure selected. If a physical structure is selected instead of a decoupled structure, more sensors are required because of the couplings. This can be shown for the same submodel combination (1, 3). For submodel 1, the output signal  $T_1$  is coupled to the signals  $T_4$ ,  $T_5$ , and  $T_7$  (eq. (7) on page 51). For submodel 3, the output signal  $T_3$  is coupled to the signals  $T_2$ ,  $T_6$  and  $T_8$  (eq. (8)). In this case, a total of eight signals ( $T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8$ ) is required for two output signals.

Using a decoupled model structure, all output signals are only coupled to two input signals and the step input, which is not measured (matrix (3) on page 49). For the two output signals  $T_1$  and  $T_3$ , only the measured input signals  $T_7$  and  $T_8$  are required, which adds up to four measured signals ( $T_1, T_3, T_7, T_8$ ). On the other hand, the rate of wrong classifications will be higher with the decoupled model structure. The sensitivities of particular parameters to the faults will be lower, since the physical relations between the parameters and the faults are almost lost. Because of higher model uncertainties, the variations of the parameters will be higher as well.

### 5.5.2 Reduction of Model Structure

Another method to reduce the number of sensors is based on the reduction of the model structure for a given subset of signals. The result is a substructure which only contains the selected output signals and input signals. During the reduction, all output signals which are not in the set of the selected output signals are eliminated. Thus, the number of required signals can be reduced more effectively for all types of model structures. But the system matrix itself changes as a result of the reduction, which means that the parameters have to be re-identified for each output signal selection and stored to a different parameter matrix. Therefore, finding the best model substructure by comparing different subsets of signals has not been automated. But in the training software tool, the reduction algorithm described below has been integrated, which allows the user to select a subset of the signals. The structure is then reduced automatically. It is recommended to start with a low number of signals and then to increase the number of signals subsequently, until the results are satisfactory, as shown in application IV of Section 6.3.

For the full model structure of matrix (4), the reduction for the output signal selection ( $T_1, T_3$ ) is straightforward in that the rows and columns  $i = (2, 4, 5, 6)$  can be eliminated, which leads to matrix (38) with four required signals ( $T_1, T_3, T_7, T_8$ ).

$$\begin{array}{c|cc|cc}
 & T_1 & T_3 & T_7 & T_8 & u_{step} \\
 \hline
 T_1 & \times & \bullet & \bullet & \bullet & \bullet \\
 T_3 & \bullet & \times & \bullet & \bullet & \bullet
 \end{array} \quad (38)$$

Using the physical model structure of matrix (2), the reduction is more delicate because of the non-uniform signal couplings. For the output signal selection ( $T_1, T_3$ ), the rows and columns  $i = (2, 4, 5, 6)$  have to be eliminated sequentially. Each eliminated output signal  $T_i = (T_2, T_4, T_5, T_6)$  is replaced by its static model equation in order to keep the physical signal couplings. For the first eliminated output signal  $T_2$ , the static model equation is given by eq. (39), according to row  $i = 2$  of the matrix (2). Thus, signal  $T_2$  depends on signal  $T_1$ .

$$T_2(t) = \begin{bmatrix} k_{2,1} & k_{2,step} \end{bmatrix} \cdot \begin{bmatrix} T_1(t) \\ u_{step}(t) \end{bmatrix} \quad (39)$$

Now, all the remaining rows of matrix (2) which contain couplings to signal  $T_2$  (rows  $i = (3, 6)$  with ‘•’ in column  $j = 2$ ) have to be modified by adding a

coupling to signal  $T_1$  (adding a ‘•’ in column  $j = 1$ ). Next, the row  $i = 2$  and the column  $j = 2$  are eliminated, which leads to matrix (40).

	$T_1$	$T_3$	$T_4$	$T_5$	$T_6$	$T_7$	$T_8$	$u_{step}$
$T_1$	×	0	•	•	0	•	0	•
$T_3$	•	×	0	0	•	0	•	•
$T_4$	0	•	×	0	0	0	0	•
$T_5$	•	0	•	×	0	•	0	•
$T_6$	•	•	0	0	×	0	•	•

(40)

This elimination and replacing procedure is repeated for the other output signals  $T_i = (T_4, T_5, T_6)$ , which yields matrix (41). If only a few output signals are left, the remaining structure obviously tends to the full model structure (cf. matrix (38)).

	$T_1$	$T_3$	$T_7$	$T_8$	$u_{step}$
$T_1$	×	•	•	0	•
$T_3$	•	×	0	•	•

(41)

For the decoupled model structure of matrix (3), this reduction algorithm would lead to the same results as those described in Section 5.5.1, since there are no couplings among the output signals that would have to be replaced.

## 5.6 Gradual and Simultaneous Faults

For handling gradual faults (faults of different sizes) or simultaneous faults (combinations of two or more faults), the clustering algorithms HCM2Fuzzy, HCM2Neuro, FCM2Neuro, or PFCM2Neuro of Section 5.4 can also be applied. In the following, the corresponding clusters will be referred to as ‘standard clusters’. Since the standard clusters are designed for discrete fault sizes, training data sets have to be generated for a multitude of different fault sizes as well as for the fault combinations (cf. Section 5.6.1). If a simulation model is available (cf. application III in Section 6.2), then the large number of data sets is generated by simulation. But if no simulation model is available, the training data sets have to be acquired on a test bench, which is time consuming for a large number of data sets. In order to reduce the number of training data sets for gradual faults, vector clusters are built (cf. Section 5.6.2).

### 5.6.1 Standard Clusters

For each trained fault size and for each trained fault combination, one cluster is built (Fig. 23). With an increasing fault size, the cluster centers are moving on a (generally nonlinear) trajectory through the parameter space, while the statistical properties (standard deviations  $\sigma_{i,k}$  of eqs. (25) and (26)) might change. The origin of all trajectories is the center of the nominal cluster. In the illustration of Fig. 23, the clusters are built for the nominal case, three discrete fault sizes (1..3) for the faults 1 and 2, as well as for one combination of the faults 1 and 2.

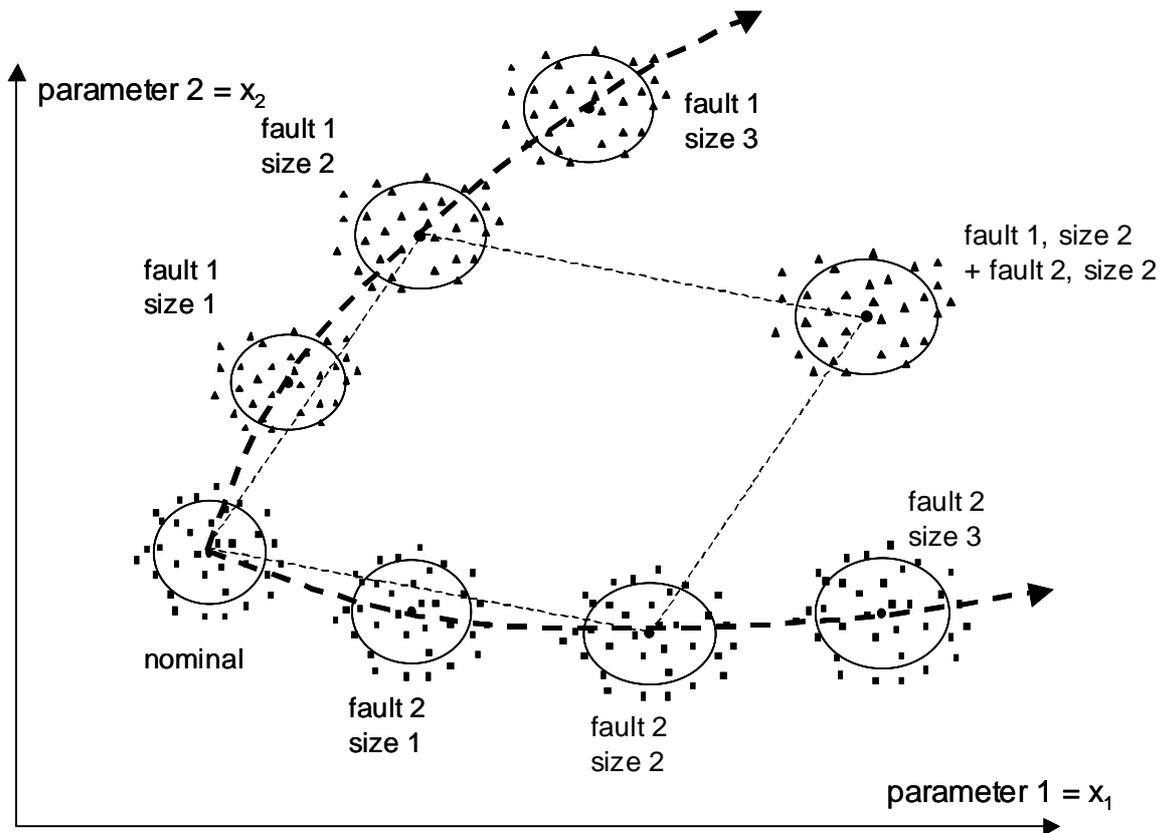


Fig. 23: Clusters for the nominal case, different fault sizes and a fault combination of two faults (illustration with two parameters)

Fig. 24 shows some *HCM2Neuro* clusters (cf. Section 5.4.2), which are built from training data in application III of Section 6.2. Here a subset of four fault cases is displayed, with four different fault sizes each.

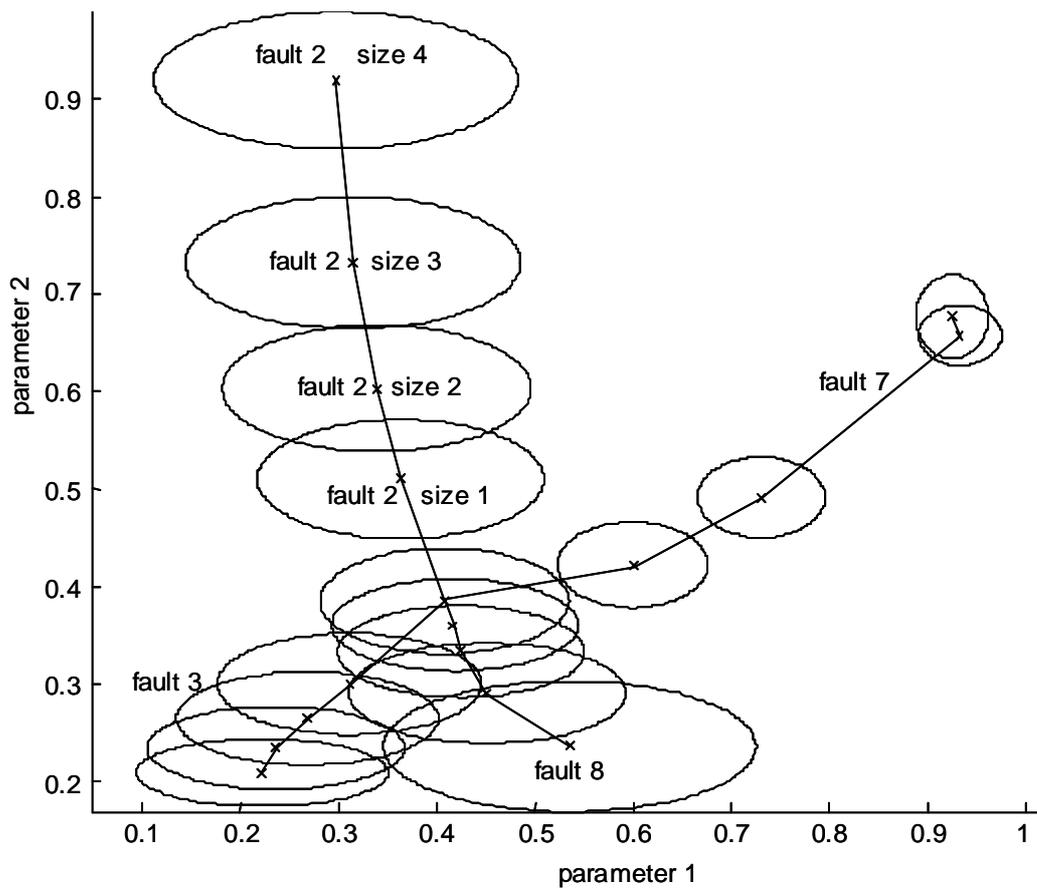


Fig. 24: Neuro cluster centers and standard deviations (ellipses) for four faults of four sizes each, projection onto two dimensions.

### 5.6.2 Vector Clusters

In order to reduce the number of training data sets for gradual faults, vector clusters are built. The vector clusters shown in Fig. 25 are created by using the centers of the nominal cluster as well as the centers of the fault clusters of one fault size each (size 2 for faults 1 and 2). As many vector clusters are built as there are different fault cases, and the trajectories of the cluster centers are assumed to be almost linear.

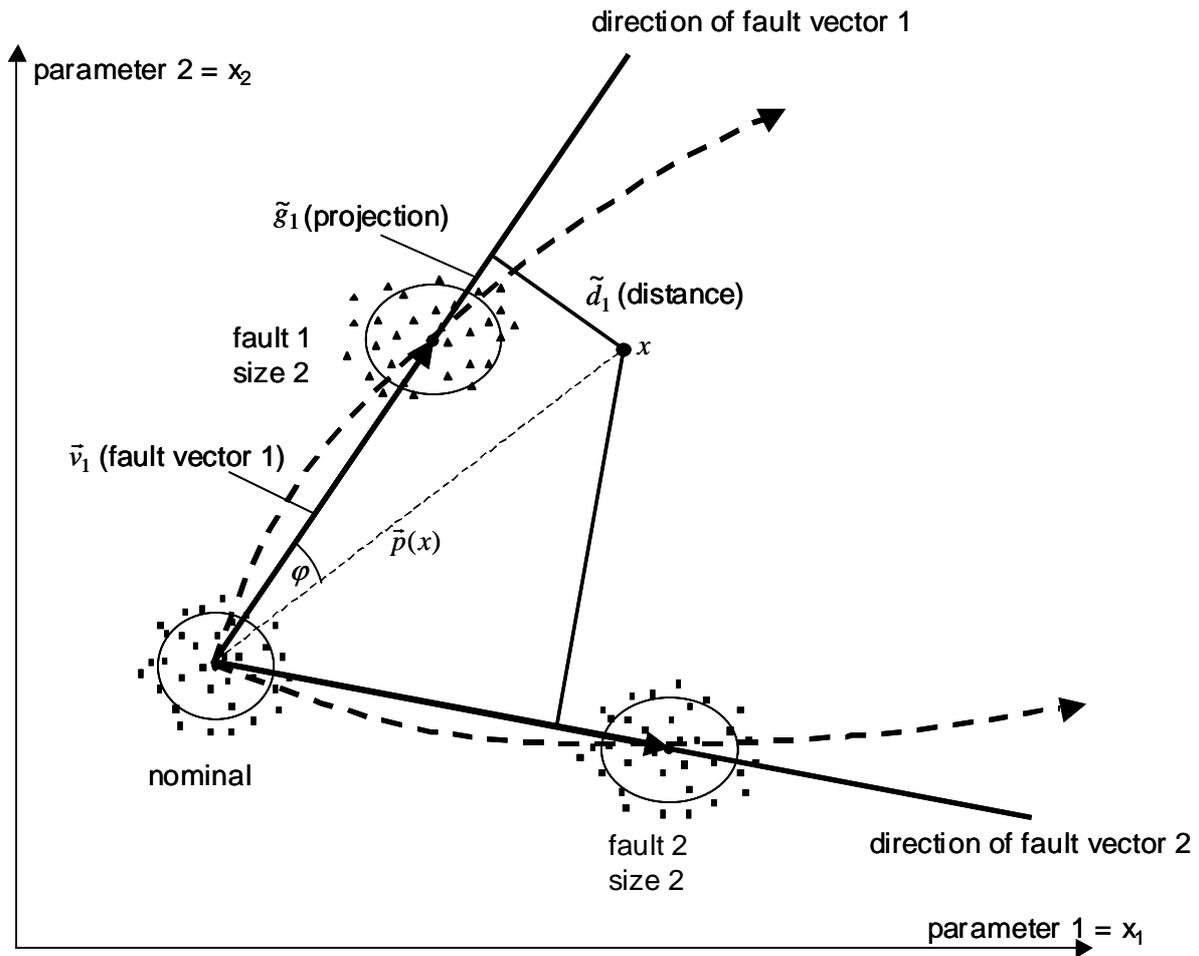


Fig. 25: Vector clusters for two faults, created from the center of the nominal cluster and the centers of the fault clusters of fault size 2 (illustration with two parameters)

During the training phase, each fault vector  $\vec{v}_i$  is defined by eq. (42), with the center  $\vec{c}_i$  of fault cluster  $i$  and the center  $\vec{c}_{nom}$  of the nominal cluster. Thus, the length  $v_i$  of the fault vector is defined by the size of the trained fault  $i$ .

$$\vec{v}_i = \vec{c}_i - \vec{c}_{nom} \quad v_i = |\vec{v}_i| = \sqrt{\sum_k v_{i,k}^2} \quad (42)$$

During operation, the fault grade  $\tilde{g}_i(x)$  of a data point  $x$  is calculated by projecting  $\vec{p}(x)$  onto the fault vector  $\vec{v}_i$  (eq. (43)). The fault grade is then scaled by the length  $v_i$  to  $g_i(x)$ .

$$\tilde{g}_i(x) = \vec{p}(x) \cdot \vec{v}_i^0 = p \cdot \cos(\varphi_i) \quad \vec{v}_i^0 = \frac{\vec{v}_i}{v_i} \quad g_i(x) = \frac{\tilde{g}_i(x)}{v_i} \quad (43)$$

The scaled fault grade  $g_i(x)$  is equal to one if the actual fault size is equal to the trained fault size ( $\tilde{g}_i(x)=v_i$ ). The distance  $\tilde{d}_i(x)$  to the fault vector  $i$  is calculated by eq. (44).

$$\tilde{d}_i(x) = \sqrt{p^2(x) - \tilde{g}_i^2(x)} \quad (44)$$

For diagnosis, the relative length of the distances  $d_i(x)$  is more important than their absolute length  $\tilde{d}_i(x)$ . Therefore, they are scaled by the minimum and maximum of the distances to all fault vectors (eq. (45)). The membership grades  $f_i(x)$  are given by linear functions of the relative distances  $d_i(x)$ . They are in the range of [0..1]. A membership grade  $f_i(x)$  close to one means a high probability that fault  $i$  is present (distance to the fault vector  $i$  is close to zero).

$$d_i(x) = \frac{\tilde{d}_i(x) - \min(\tilde{d}_i(x))}{\max(\tilde{d}_i(x)) - \min(\tilde{d}_i(x))} \quad f_i(x) = 1 - d_i(x) \quad (45)$$

With the vector clusters, the training effort is reduced to the nominal case and to one size for each fault. The resulting classification quality is verified by validation data of different fault sizes in the applications of Sections 6.2 and 6.3.

## 5.7 Online Fault Diagnosis

### 5.7.1 Diagnosis Steps

The online fault diagnosis system applies the trained classification system and executes the following steps (cf. online software in Section 7.2):

- *Online data logging and preprocessing.* The online data is acquired and stored for each data sequence (cf. Fig. 10 on page 45 for an example off-on sequence). The data is cut to a constant sequence length.
- *Parameter identification.* For the selected model structure (cf. Section 5.2) and the selected model approaches (static, 1<sup>st</sup> order ARX, ...; cf. Section 5.3.3), the parameters of the submodels are identified using the data of the current sequence (cf. Section 5.3). The identified parameters are scaled and stored in one parameter set for the current data sequence.

- *Fault classification.* With the selected cluster type (cf. Section 5.4) and the trained cluster properties (centers, standard deviations, etc.), the current parameter set is classified. A membership grade is calculated for each fault case. In the case of vector clusters, the fault grades are also calculated.
- *Statistical Evaluation.* In order to significantly improve the classification results, the results of several sequences are averaged (cf. Section 5.7.2).

Whereas for the training phase additional sensors are mounted on the test bench for finding the best qualified signals, the online software only requires the sensors corresponding to the selected signals.

### 5.7.2 Averaging

Each data sequence  $i_{seq}$  yields one parameter set  $x(i_{seq})$ , which is classified to the membership grade vector  $f(x(i_{seq}))$  (eq. (46)) containing the grades  $f_i(x(i_{seq}))$  for each fault cluster  $i=1..m$  (cf. Fig. 16 on page 64).

$$f(x(i_{seq})) = [f_1(x(i_{seq})) \quad f_2(x(i_{seq})) \quad \cdots \quad f_m(x(i_{seq}))]^T \quad (46)$$

Several data sequences are combined in one cycle  $i_{cyc}$ . At the end of each cycle, the grades are statistically evaluated. For averaging, eq. (47) or eq. (48) can be used. Eq. (47) calculates the mean values of the membership grades over  $l$  sequences, *after* classification.

$$\bar{f}(i_{cyc}) = \frac{1}{l} \sum_{s=1}^l f(x(s)) \quad (47)$$

Eq. (48) *first* calculates the parameter mean values and then classifies the mean values by their membership grades. Typically, the parameters of about  $l=5$  sequences are averaged.

$$\bar{f}(i_{cyc}) = f\left(\frac{1}{l} \sum_{s=1}^l x(s)\right) \quad (48)$$

Here, eq. (48) is applied, because it is more robust for the case of outliers or high variances, using nonlinear membership functions  $f(x)$ . For example, an outlier could be represented by a data point  $x$  belonging to fault case  $i$ , which is very close to a center of another fault cluster  $j$ . With a nonlinear (Gaussian) membership function  $f_j(x)$ , the membership grade to the wrong fault cluster  $j$  would be dominant. In the case of high variances, similar considerations can be made.

Using vector clusters, the same statistical evaluation is made for the fault grades  $g_i(x)$ . Averaging increases the robustness of the classification system and decreases the uncertainty (uncertainty  $\sim 1/\sqrt{l}$  with  $l$  representing the number of averaged sequences). Thus the number of wrong classifications is almost reduced to zero, which is shown in the results presented in Chapter 6.3. The mean values are visualized by the software tool described in Sections 7.2.2 and 7.2.3.

### 5.7.3 Fault Trend Charts

The cycle mean values of the membership grades (and fault grades) are plotted in trend charts (Fig. 26). These trend charts are informative for the progression of the faults. With the knowledge of the actual fault situation, a condition-oriented maintenance is possible. The trend charts are also visualized by the software tool in Section 7.2.4.

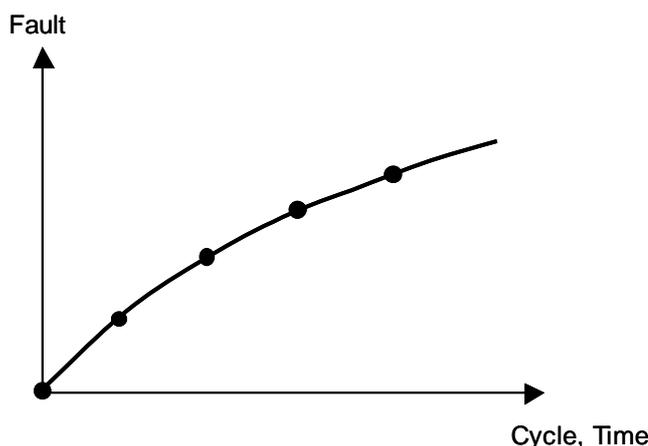


Fig. 26: Trend chart for one fault over time, dots = mean values of each cycle.

### 5.7.4 False Alarm Tuning

During the training procedure, the rate of wrong classifications is checked by the algorithm of Section 5.4.4, and it is minimized by selecting the appropriate signals, submodel approaches, and cluster types. For online operation, not only a minimal rate of wrong classifications, but also a minimal false alarm rate is mandatory. A false alarm occurs if the diagnosis system displays a fault, even if in reality there is no fault (cf. definition in eq. (33)). In order to reduce the false alarm rate for a trained classification system, a tuning parameter is introduced.

For the standard clusters of Section 5.4, the membership grades of the nominal cluster  $f_{nom}(x)$  are weighted with a factor  $w_{nom} \geq 1$  (eq. (49)).

$$f_{nom}^*(x) = w_{nom} \cdot f_{nom}(x) \quad (49)$$

Fig. 27 shows the effect of the nominal weighting. The higher the weighting factor is chosen, the lower the false alarm rate is. If the weighting factor is too high, hardly any faults will be detected anymore and the missed detection rate will increase (cf. definition of a missed detection in eq. (34)). Finding the right trade-off is at the option of the manufacturer. Both the false alarm rate and the missed detection rate influence the wrong classification rate of eq. (35). On the other hand, minimizing the wrong classification rate will not automatically minimize the false alarm rate, because the wrong classification rate describes the wrong classifications among *all* clusters, whereas the false alarm rate only describes the wrong classifications among the *nominal* cluster and its neighbours. Nominal weighting has been used in application IV, Section 6.3.8,

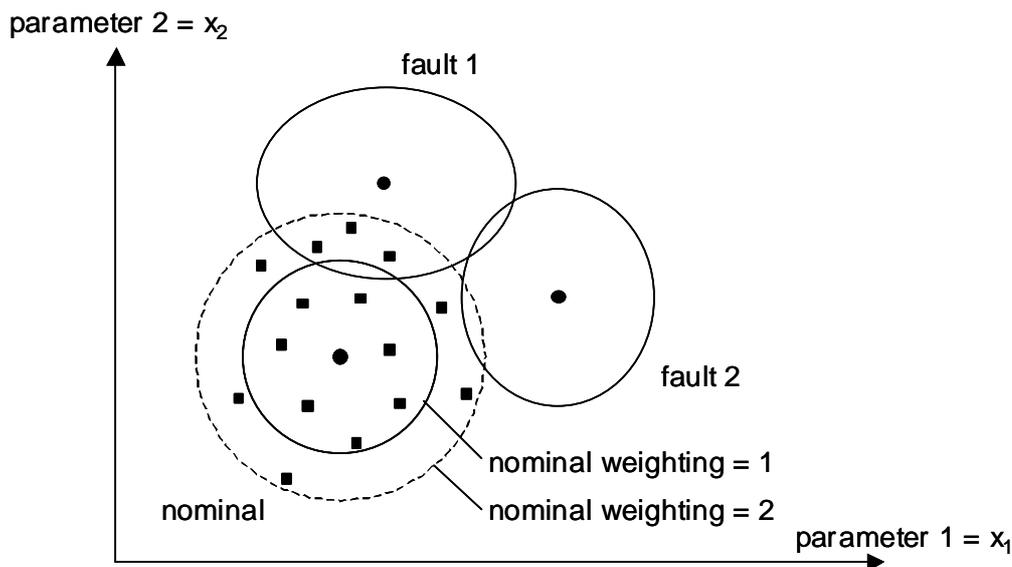


Fig. 27: Nominal weighting for standard clusters (illustrative example). The membership grades of the nominal cluster are weighted with the factor 1 or 2. With factor 1, three nominal points (rectangles) belong to fault cluster 1 instead of the nominal cluster. With factor 2, all nominal points (rectangles) belong to the nominal cluster.

For the vector clusters of Section 5.6.2, a minimum fault grade  $g_{min}$  is introduced (eq. (50), Fig. 28). If the fault grade  $g_i(x)$  of a data point  $x$  is smaller than  $g_{min}$  for a fault vector  $i$ , then the membership grade  $f_i^*(x)$  is set to zero. If the fault grades  $g_i(x)$  are smaller than  $g_{min}$  for all fault vectors  $i$ , then the data point  $x$  is classified to the nominal case.

$$f_i^*(x) = \begin{cases} f_i(x) & g_i(x) \geq g_{min} \\ 0 & g_i(x) < g_{min} \end{cases} \quad (50)$$

The minimum fault grade  $g_{min}$  has a similar effect as the nominal weighting factor above. The same trade-off between a minimal false alarm rate and a minimal missed detection rate has to be made by choosing an appropriate value for  $g_{min}$ .

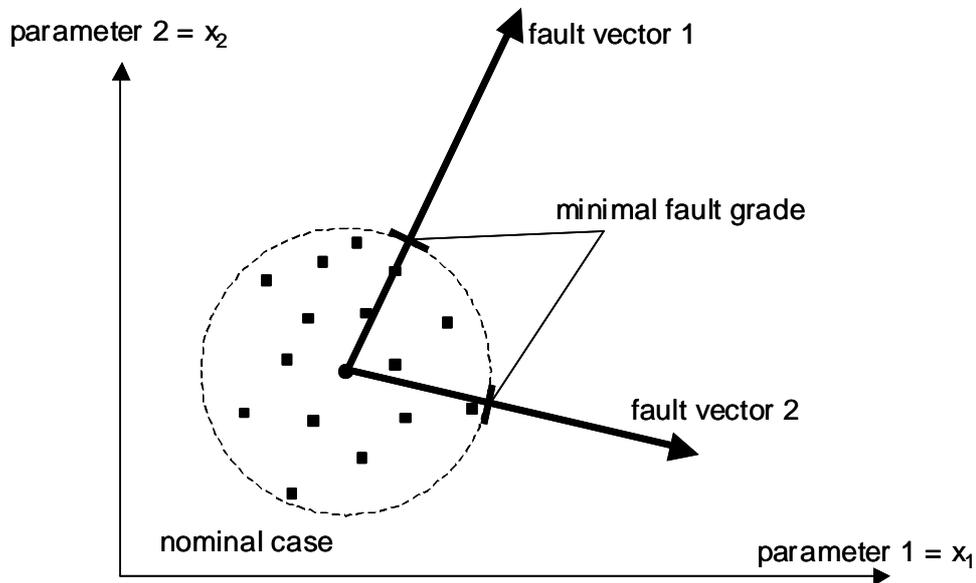


Fig. 28: Minimal fault grade for vector clusters (illustrative example). With an appropriate minimal fault grade, all nominal points (rectangles) are classified to the nominal case.

The false alarm rate could be reduced to zero in application IV, Section 6.3.8, with  $g_{min} = 0.37$  for the given validation data set, without significantly increasing the missed detection rate, which is shown by the low wrong classification rate (fault grade  $g = 1$  for trained fault size).

## 6 A Fault Diagnosis System with Integrated Fault Classification – Applications

The training methods of Chapter 5 have been integrated into a software tool, which has been applied to several measured data sets. Data was acquired from a laboratory heat pump of the Zurich University of Applied Sciences, Winterthur (ZHW), from the test bench at the Measurement and Control Lab (IMRT), and the simulation model. An application overview is given by Table 1 on page 29 (column *FuzzyWatch*).

The intention of this chapter is to demonstrate that the same training methods are generally applicable for data sets from different types of heat pumps. The signal plots as well as the model error charts, the parameter charts and the cluster plots in the subsequent sections are produced by the software tool. They allow a visualization of the training steps, which are automated. With these visualizations, the user is able to supervise the training process. Still, the required user inputs are reduced to a minimum for a maximum usability.

The classification results, expressed in the rate of wrong classifications, are given for each application. They are calculated for comparing different methods, such as different model approaches or clustering methods, within one application. But they are not intended to be used for comparing different applications, since they depend on the type and the size of the trained faults as well as on the configuration of the sensors. Nevertheless, the aim is to keep the wrong classification rates lower than 10%. Using the averaging method of Section 5.7.2, the results can be additionally improved. This method is applied in Section 6.3 only.

Chapter 7 describes one version of the software tool, which has been adapted to the heat pump of the IMRT test bench. This application IV will first be presented in Section 6.3. Application I (with Barzheim data) is not used here. It will be used for testing the other diagnosis system (cf. Section 9.1)

### 6.1 Data from ZHW Test Bench (Application II)

In this section, various signal plots are shown for visualizing the measured data sequences and for showing the fitting quality of the corresponding models. But the signals do not have to be interpreted by the user, since the training methods are doing this automatically by a statistical evaluation of the corresponding model parameters. After presenting the model error charts and pa-

parameter charts, various clustering methods will be tested for the training of the fault classification [Zogg 01/1].

### 6.1.1 Configuration of the Test Heat Pump

The laboratory brine-to-water heat pump contains an additional subcooler (Fig. 29). Here, the existing actuators and sensors of the heat pump are used to set the faults and measure the signals. No further modifications to the configuration are made. The data sheet of the heat pump is given in Appendix C, Section ‘ZHW Laboratory Heat Pump’.

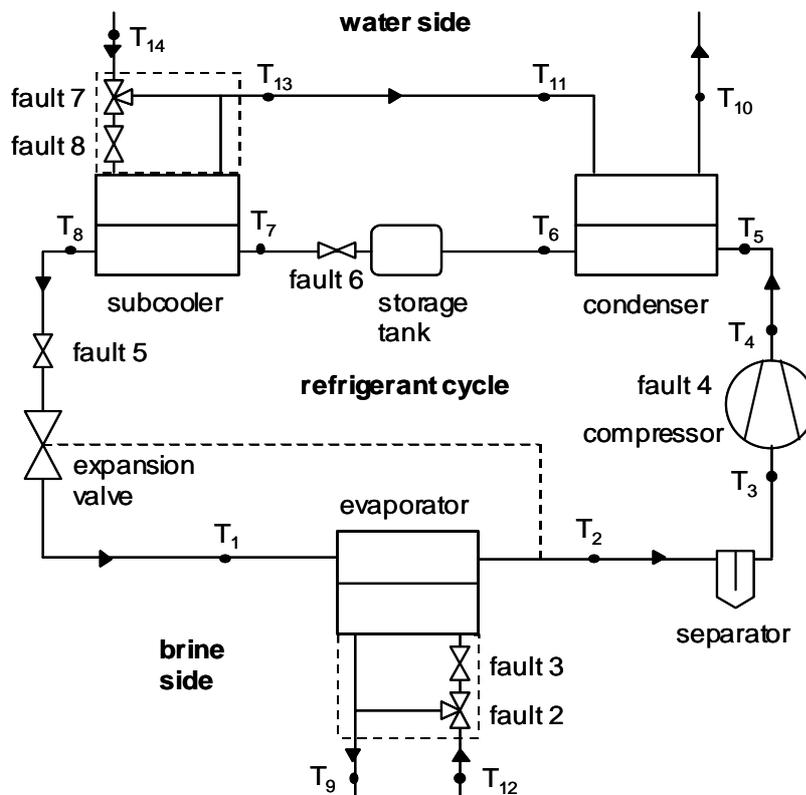


Fig. 29: The laboratory brine-to-water ZHW heat pump, with additional valves (faults 2,3,5,6,7,8) and temperature sensors ( $T_1..T_{14}$ ).

The manual 2-way and 3-way valves affect the refrigerant cycle, the brine cycle, and the water cycle. The temperature of the media is measured by a total of 13 sensors. They are used as an initial sensor configuration, since during the training, the number of sensors will be reduced. The temperatures  $T_{11}$  and  $T_{13}$  are measured by the same sensor and therefore are only counted once ( $T_{11}=T_{13}$ ). A sampling time of 5 seconds has been chosen, whereas the data sequences have a duration of 15 minutes.

All faults are introduced by adjusting the corresponding valves, except for fault 4, which is set by the compressor speed (cf. Table 3). By opening the 3-way valve of fault 2, the brine mass flow is separated into a part that flows through the bypass and a part that flows through the evaporator. Since the remaining part of the brine mass flow through the evaporator is smaller, the brine is cooled down to a lower temperature level. Since the temperature difference between the brine side and the refrigerant side in the evaporator is smaller, the heat transfer is reduced. The cooled part is then mixed with the warmer part that flows through the bypass. The effect of opening the bypass is similar to that of reducing the heat transfer coefficient due to fouling. Analogously, fault 7 affects the heat transfer at the subcooler. All faults are described in Table 3 (cf. Table 2 on page 41 also). For each fault case, 15 data sequences were measured.

Table 3: The faults of the ZHW heat pump

fault	actuator	fault description
nominal (fault 1)	-	no fault
fault 2	3-way valve (bypass)	reduced evaporator heat transfer (evaporator fouling on brine side)
fault 3	valve	reduced evaporator brine mass flow (fouling on brine side, brine pump malfunction)
fault 4	compressor speed	reduced compressor mass flow (compressor malfunction)
fault 5	valve	reduced expansion valve mass flow (refrigerant line restriction, expansion valve malfunction)
fault 6	valve	reduced subcooler mass flow (refrigerant line restriction)
fault 7	3-way valve (bypass)	reduced subcooler heat transfer (heat exchanger fouling on water side)
fault 8	valve	reduced subcooler water mass flow (fouling on water side, water pump malfunction)

### 6.1.2 Model

The input vector  $u(t)$  and the output vector  $y(t)$  are defined in eq. (51), ( $T_{11}=T_{13}$ ).

$$y(t) = \begin{bmatrix} T_1(t) \\ T_2(t) \\ \vdots \\ T_{11}(t) \end{bmatrix} \quad u(t) = \begin{bmatrix} T_{12}(t) \\ T_{14}(t) \\ u_{step}(t) \end{bmatrix} \quad (51)$$

The system matrix of the physical model structure is given by matrix (52), corresponding to Fig. 29. It contains 11 submodels, corresponding to each row of the matrix.

	$T_1$	$T_2$	$T_3$	$T_4$	$T_5$	$T_6$	$T_7$	$T_8$	$T_9$	$T_{10}$	$T_{11}$	$T_{12}$	$T_{14}$	$u_{step}$
$T_1$	×	0	0	0	0	0	0	•	0	0	0	0	0	•
$T_2$	•	×	0	0	0	0	0	0	•	0	0	•	0	•
$T_3$	0	•	×	0	0	0	0	0	0	0	0	0	0	•
$T_4$	0	0	•	×	0	0	0	0	0	0	0	0	0	•
$T_5$	0	0	0	•	×	0	0	0	0	0	0	0	0	•
$T_6$	0	0	0	0	•	×	0	0	0	•	•	0	0	•
$T_7$	0	0	0	0	0	•	×	0	0	0	0	0	0	•
$T_8$	0	0	0	0	0	0	•	×	0	0	•	0	•	•
$T_9$	•	•	0	0	0	0	0	0	×	0	0	•	0	•
$T_{10}$	0	0	0	0	•	•	0	0	0	×	•	0	0	•
$T_{11}$	0	0	0	0	0	0	•	•	0	0	×	0	•	•

(52)

The other model structures are in analogy to those described in Section 5.2. The physical as well as the decoupled model structures are used for training. The heat pump has two modes (off, on). Here the parameters of the off-on sequences are identified and evaluated.

### 6.1.3 Signal Plots for the Physical Model Structure

After the identification of the parameters, the submodels are simulated. Fig. 30 shows the measured data and the simulated output signals for a data sequence of the nominal case (fault-free case), using the physical model structure. These plots are used for visualizing the fitting quality of the model structure selected. For each submodel, the static model approach as well as the first-order ARX approach are simulated. The signals  $T_{12}$  and  $T_{14}$  are inputs to all submodels, corresponding to the current operating point. Here, the output signals for the five submodels 1, 2, 5, 8, and 9 are shown, which are

representative for all eleven output signals. In the title of the plots for each output signal, the coupled signals are also displayed, which correspond to the structure matrix (52). The plots for the output signals  $T_2$  and  $T_8$  in Fig. 30 correspond to the plots of the output signals  $T_1$  and  $T_3$  in Fig. 13 (page 55). Only, here the numbering of the signals is different due to the special heat pump type.

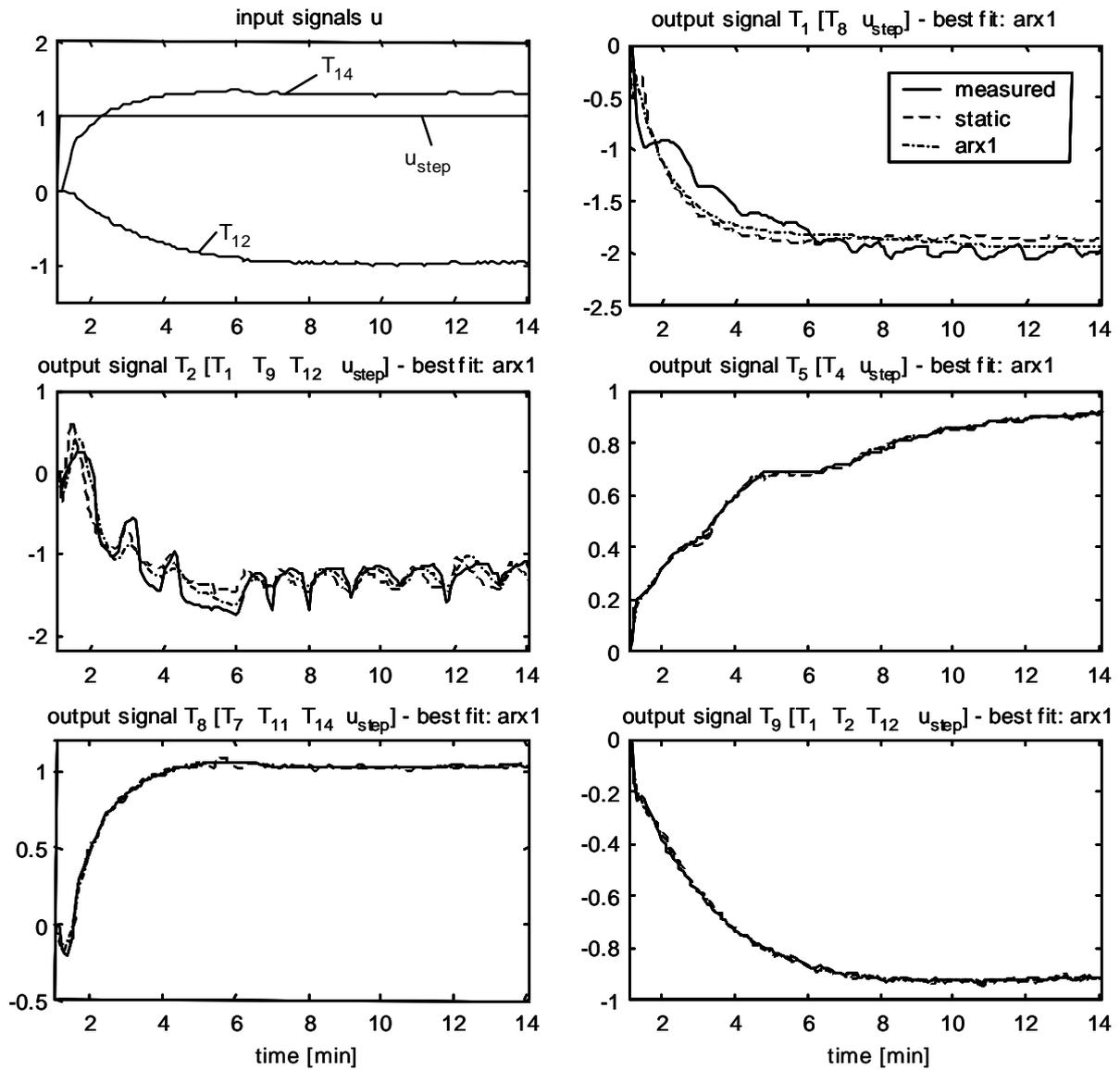


Fig. 30: Nominal data sequence I, physical model structure. Upper left plot: measured input signals ( $T_{12}$ ,  $T_{14}$ ) and step input signal  $u_{step}$ . Other plots: measured (solid line) and simulated output signals  $T_1$ ,  $T_2$ ,  $T_5$ ,  $T_8$ ,  $T_9$  for the static and first order ARX approaches. All signals are scaled. The coupled signals are listed in brackets [], for example:  $T_1$  is coupled with the signals  $T_8$  and  $u_{step}$ . The approaches which best fit the measured data of the current sequence are noted in the title of the plots.

Oscillations due to the ‘hunting effect’ are visible in the measured signal  $T_2$ , but also in the measured signal  $T_1$ . In contrast to the simulated output signal  $T_2$ , the simulated output signal  $T_1$  is not able to follow the transients and the oscillations of the measured signal, because it is only coupled to the signal  $T_8$ , which has different transients and in which the oscillations are very small. On the other hand, the simulated output signals  $T_5$ ,  $T_8$  and  $T_9$  fit the measured data very well. Similarly low errors are found in the signals  $T_6$ ,  $T_{10}$ , and  $T_{11}$ , which are not shown here. In the signal  $T_5$ , some ‘nonlinearities’ are visible, which are indirectly related to the nonuniform oscillations of  $T_2$ .

For illustrating the variations of the output signals within the data of the nominal case, Fig. 31 shows the output signals  $T_2$ ,  $T_5$  and  $T_9$  for another nominal data sequence.

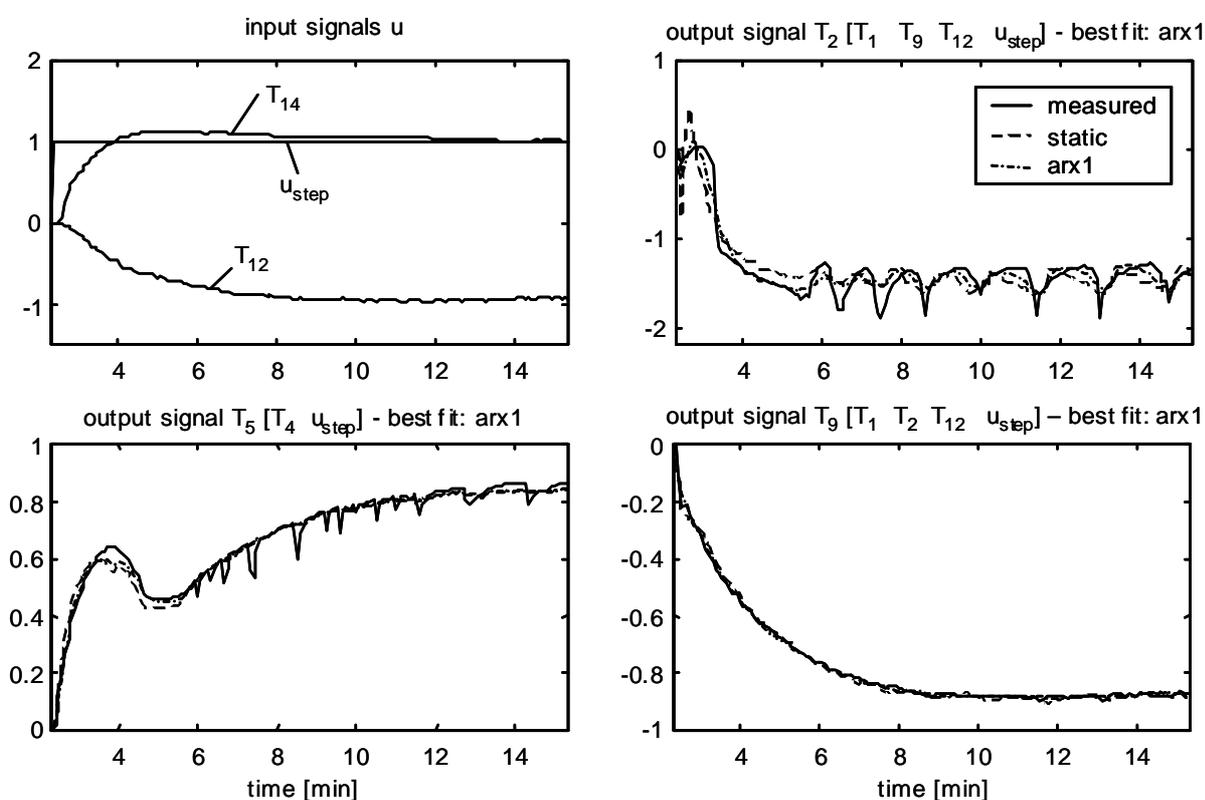


Fig. 31: Nominal data sequence II, physical model structure. Upper left plot: measured input signals ( $T_{12}$ ,  $T_{14}$ ) and step input signal  $u_{step}$ . Other plots: measured (solid line) and simulated output signals  $T_2$ ,  $T_5$  and  $T_9$ . All signals are scaled.

In particular, the transients of signal  $T_5$  have changed very much, which is indirectly related to the changed distribution of the oscillations in  $T_2$ . Obviously, here the compressor got ‘stuck’ during the startup, being influenced by the reaction of the thermostatic expansion valve (oscillations in  $T_2$  start at  $t = 4.8$  min. and  $T_5$  increases again). This effect was observed, to varying degrees, in about 50% of the data sequences of the nominal case and of the fault

cases, but it was not related to any particular fault case. Thus, it is treated as a ‘stochastic’ variation of the signals. The other output signals, such as signal  $T_9$  in Fig. 31, have much lower variations. They will be important for classifying the faults.

In contrast to the nominal data sequences above, Fig. 32 shows a data sequence of fault case 2. Comparing the output signals  $T_2$ ,  $T_5$ , and  $T_9$  in Fig. 32 to those in Fig. 30 or Fig. 31, some changes are visible. For example, the amplitude of the oscillations in the signal  $T_2$  is higher and the level of their mean value is lower in quasi steady-state. The steady-state level of the signal  $T_9$  is slightly increased. These changes are related to fault case 2, which is a reduction of the heat transfer in the evaporator. However, the direct interpretation of the signals is very difficult, partly because of the variations due to disturbances. In addition, the output signal levels discussed are influenced by the operating point, thus by the varying levels of the input signals. Therefore, the corresponding submodel parameters have to be considered, which describe the relation of the output signals to the input signals.

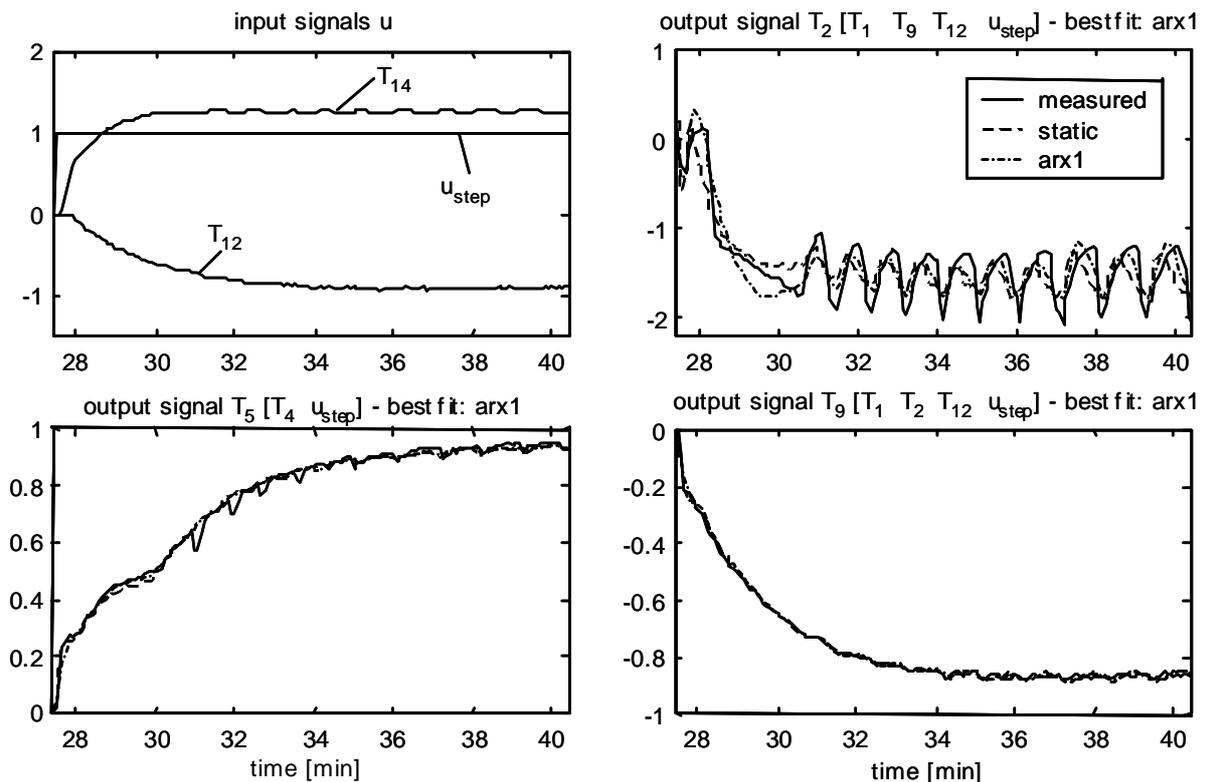


Fig. 32: Data sequence of fault 2 (reduced evaporator heat transfer), physical model structure. Upper left plot: measured input signals ( $T_{12}$ ,  $T_{14}$ ) and step input signal  $u_{\text{step}}$ . Other plots: measured (solid line) and simulated output signals  $T_2$ ,  $T_5$  and  $T_9$ . All signals are scaled.

Fig. 33 shows a data sequence of fault case 3. Comparing the same output signals to the fault cases above, some changes are visible in the steady-state level of the signal  $T_9$  for example, which is lower. A lower brine outlet temperature  $T_9$  is related to fault case 3, namely a reduced brine mass flow through the evaporator. Here also, the output signal levels are additionally depending on the input signal levels.

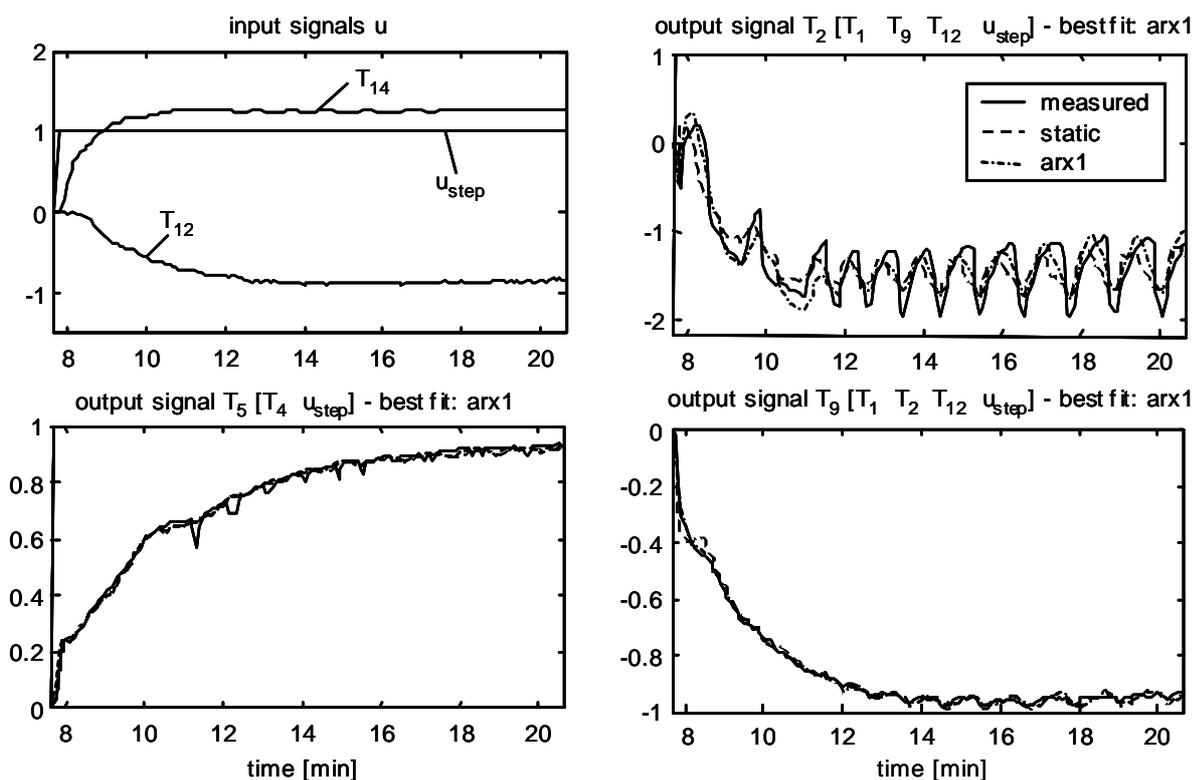


Fig. 33: Data sequence of fault 3 (reduced evaporator mass flow), physical model structure. Upper left plot: measured input signals ( $T_{12}$ ,  $T_{14}$ ) and step input signal  $u_{step}$ . Other plots: measured (solid line) and simulated output signals  $T_2$ ,  $T_5$ , and  $T_9$ . All signals are scaled.

In the discussions above, one data sequence was analyzed for each of the fault cases 2 and 3. Clearly, the interpretation of the faults is difficult and it is obvious that instead of the signals the submodel parameters have to be regarded. Additionally, no reliable interpretations can be made by regarding one data sequence per fault case only, since the data sequences vary considerably within each fault case. Therefore, the submodel parameters of 15 data sequences per fault case will be statistically evaluated by the software tool such that no interpretations have to be made by the user.

### 6.1.4 Error Charts for the Physical Model Structure

For the same output signals as in Fig. 30, the model errors (the differences between the simulated and measured output signals) are plotted in Fig. 34. But instead of only regarding one data sequence, here the model errors are statistically evaluated over 15 data sequences for each fault case 1..8 (fault 1 = nominal case).

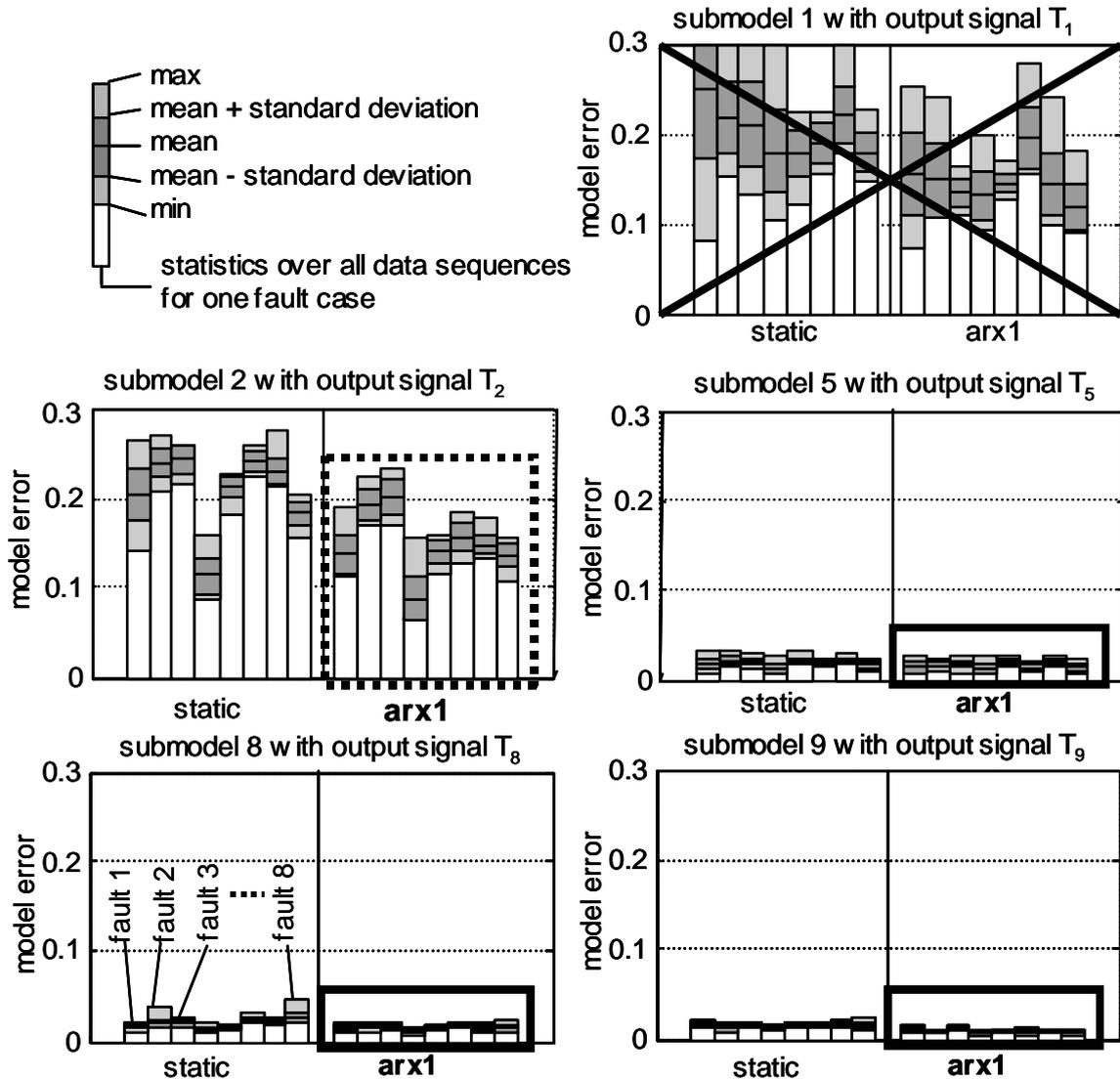


Fig. 34: Error charts for the submodels 1, 2, 5, 8, 9 with the output signals  $T_1$ ,  $T_2$ ,  $T_5$ ,  $T_8$ ,  $T_9$ , physical model structure. For each plot, the errors of the static model are in the left bar group, and the errors of the 1<sup>st</sup> order ARX model are in the right bar group; within a group, each bar represents one fault case (faults 1..8 from left to right). Legend for the bars (on the upper left): light gray region = minimum/maximum values of the errors, dark gray region = standard deviations of the errors

The model errors of submodel 1 are too high, which has already been shown for the signal  $T_1$  in Fig. 30. Therefore this submodel is eliminated. The errors

of submodel 2 are not low, either. But since the oscillations and the level of the output signal  $T_2$  contains important information about the faults, it is not eliminated. Therefore, the ARX approach with lower errors is selected. The other submodels have very low errors for both the static model approaches and the ARX model approaches. Here, the ARX approaches are selected also for convenience. In the present application, these submodel selections are made by the user. As a result of the error charts above and the other error charts for the remaining submodels, the ARX approaches have been selected for the submodels 2 and 4..11, whereas the submodels 1 and 3 have been eliminated.

### 6.1.5 Parameter Charts for the Physical Model Structure

The parameter sensitivities of the selected ARX approaches are shown in Fig. 35 for the submodels 2, 5, 8, and 9. These charts lead to the conclusion that the submodels 2, 8, and 9 will be important for classifying the faults, because at least some of their parameters have rather high sensitivities in relation to the variances. Because of the higher signal variations in the output signal  $T_5$  (cf. Fig. 30 and Fig. 31), the corresponding submodel 5 generally has higher variations in its parameters and lower sensitivities. But it can also be used, at least for facilitating the classification of fault case 6 (parameters  $a_5$ ,  $b_{5,3}$ ,  $6^{\text{th}}$  bar from the left which is ‘lower’).

Initially, all preselected submodels 2, 4..11 are taken into account. Using the ARX approach for each submodel, this adds up to 39 parameters. The number of submodels is then reduced. Finding the best combination of submodels with the corresponding parameters is automated by the software tool, using the algorithm presented in Section 5.5.1.

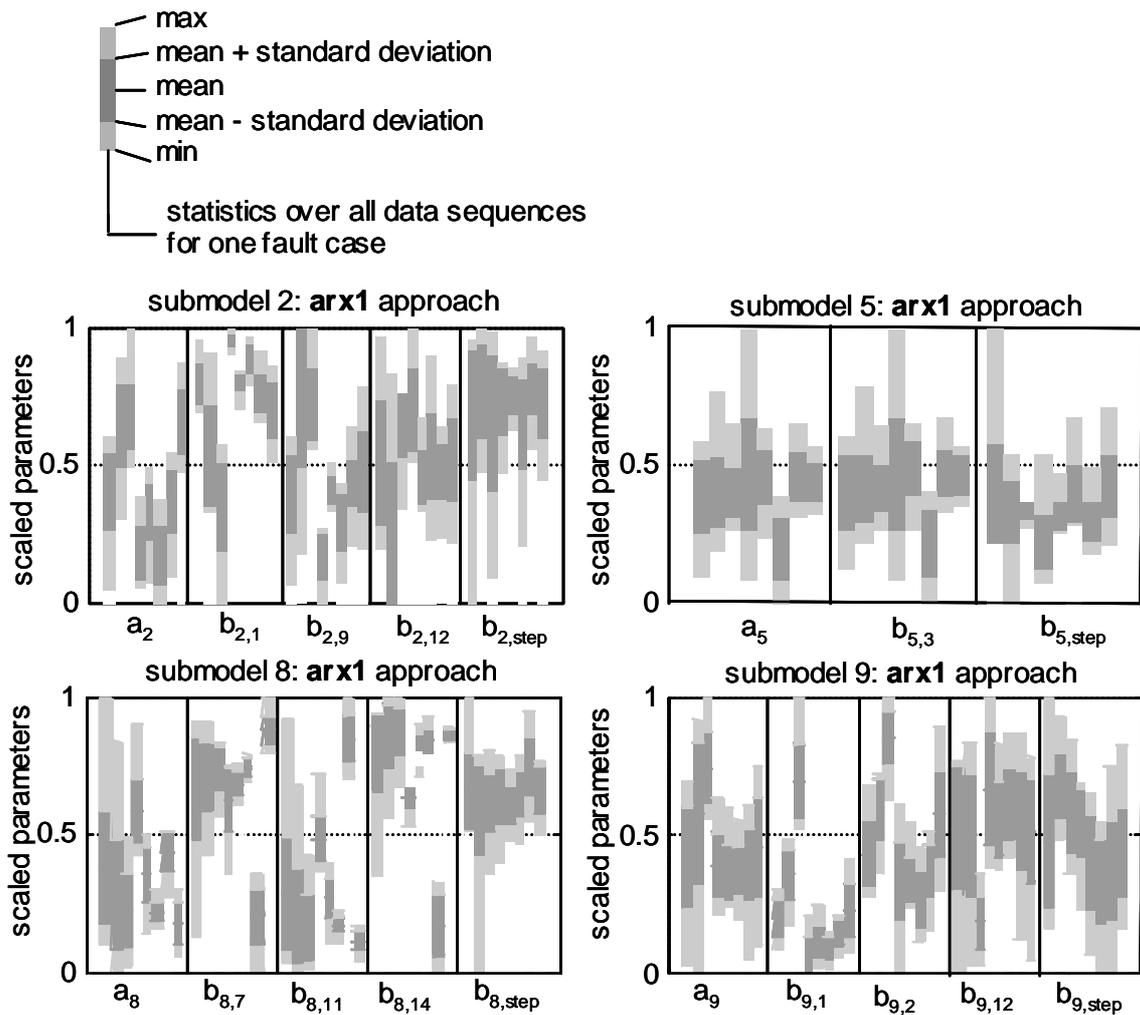


Fig. 35: Parameter charts for the submodels 2, 5, 8, 9 with the first order ARX approaches selected. Within a plot, each bar group represents one parameter (plot for submodel 2: parameters  $a_2$ ,  $b_{2,b}$  ... from left to right); within a group, each bar represents one fault case (faults 1..8 from left to right). Legend for the bars (on the upper left): light gray region = minimum/maximum parameter values, dark gray region = standard deviations of the parameters. All parameters are scaled.

### 6.1.6 Signal Plots for the Decoupled Model Structure

In order to compare different model structures, here the decoupled model structure has also been selected for training. The signal plots are shown in Fig. 36 for the same data sequence as in Fig. 30. Here all output signals are coupled to the same input signals  $T_{14}$  and  $T_{12}$ . Obviously the fitting quality of the decoupled structure differs from that of the physical structure. But the model errors are not generally larger than those of the physical structure. For the output signal  $T_I$ , the model error is even smaller, because here the transients of the coupled signals  $T_{12}$  and  $T_{14}$  are closer to the output signal  $T_I$  than

the transients of the coupled signal  $T_8$  in the case of the physical structure. The output signal  $T_2$  fits worse than in Fig. 30, since here the oscillations are only very small in the coupled input signals. Because of unsuitable couplings, the reconstruction of the output signal  $T_5$  also is more difficult. But the errors of the output signals  $T_8$  and especially  $T_9$  are low. These signals are highly correlated to the input signals  $T_{12}$  and  $T_{14}$ , which fulfills the assumption of the decoupled model structure. Similarly low errors are found in the signals  $T_6$ ,  $T_7$ ,  $T_{10}$ , and  $T_{11}$ , which are not shown here. Thus, the decoupled model structure will also be used for a further training.

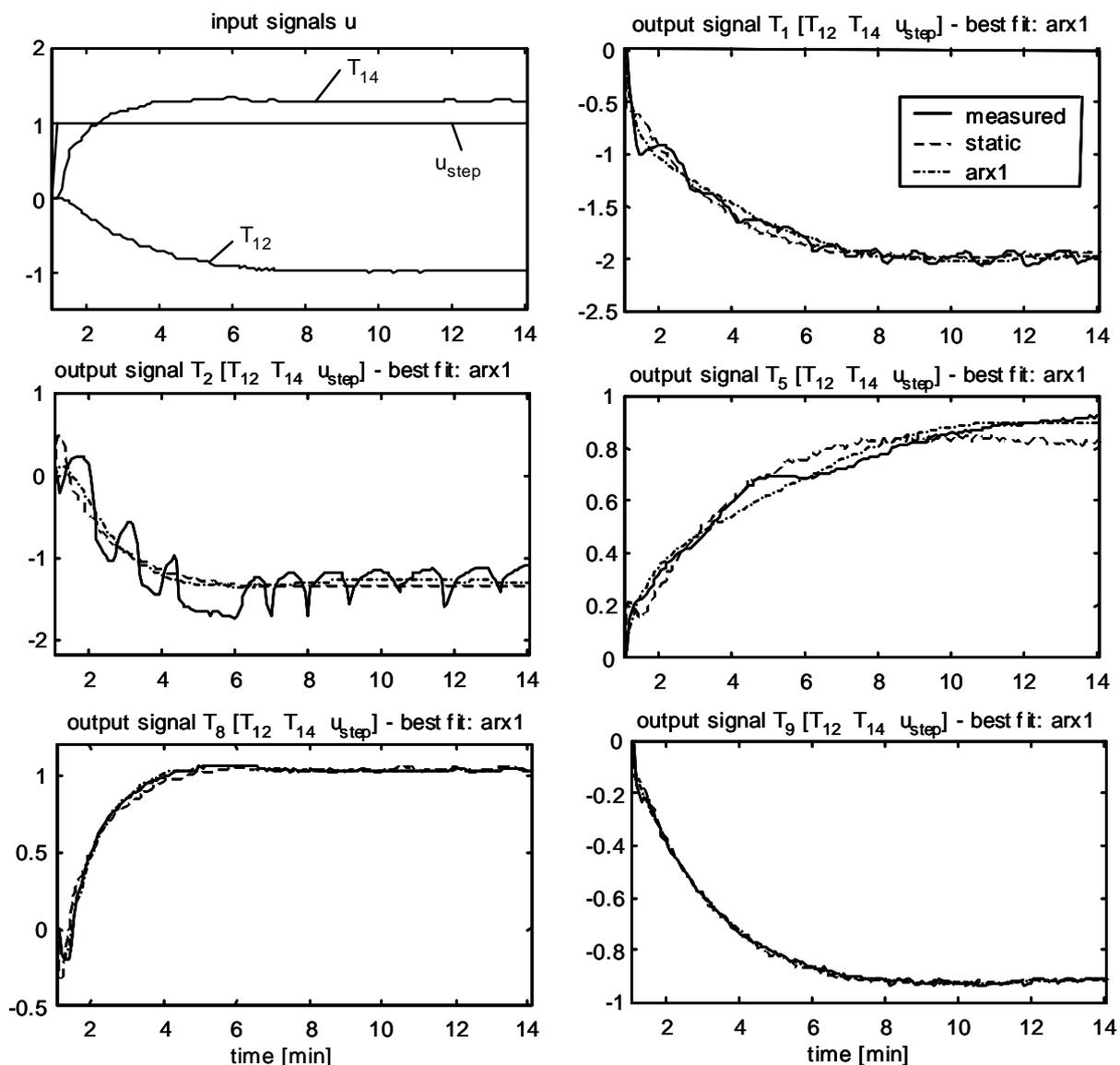


Fig. 36: Nominal data sequence I, decoupled model structure. Upper left plot: measured input signals ( $T_{12}$ ,  $T_{14}$ ) and step input signal  $u_{step}$ . Other plots: measured (solid line) and simulated output signals  $T_1$ ,  $T_2$ ,  $T_5$ ,  $T_8$ ,  $T_9$  for the static and first order ARX approaches. All signals are scaled.

### 6.1.7 Error Charts for the Decoupled Model Structure

The model errors are plotted in Fig. 34 which are statistically evaluated over 15 data sequences for each fault case 1..8 (fault 1 = nominal case). Here, submodel 1 is selected in contrast to the physical structure in Fig. 34, but submodel 2 is eliminated. As expected, the errors of submodel 5 are higher than they were with the physical structure, but they are still at a low level. For the submodels 8 and 9, the errors are very low. Because of generally lower model errors, the ARX approaches have been selected.

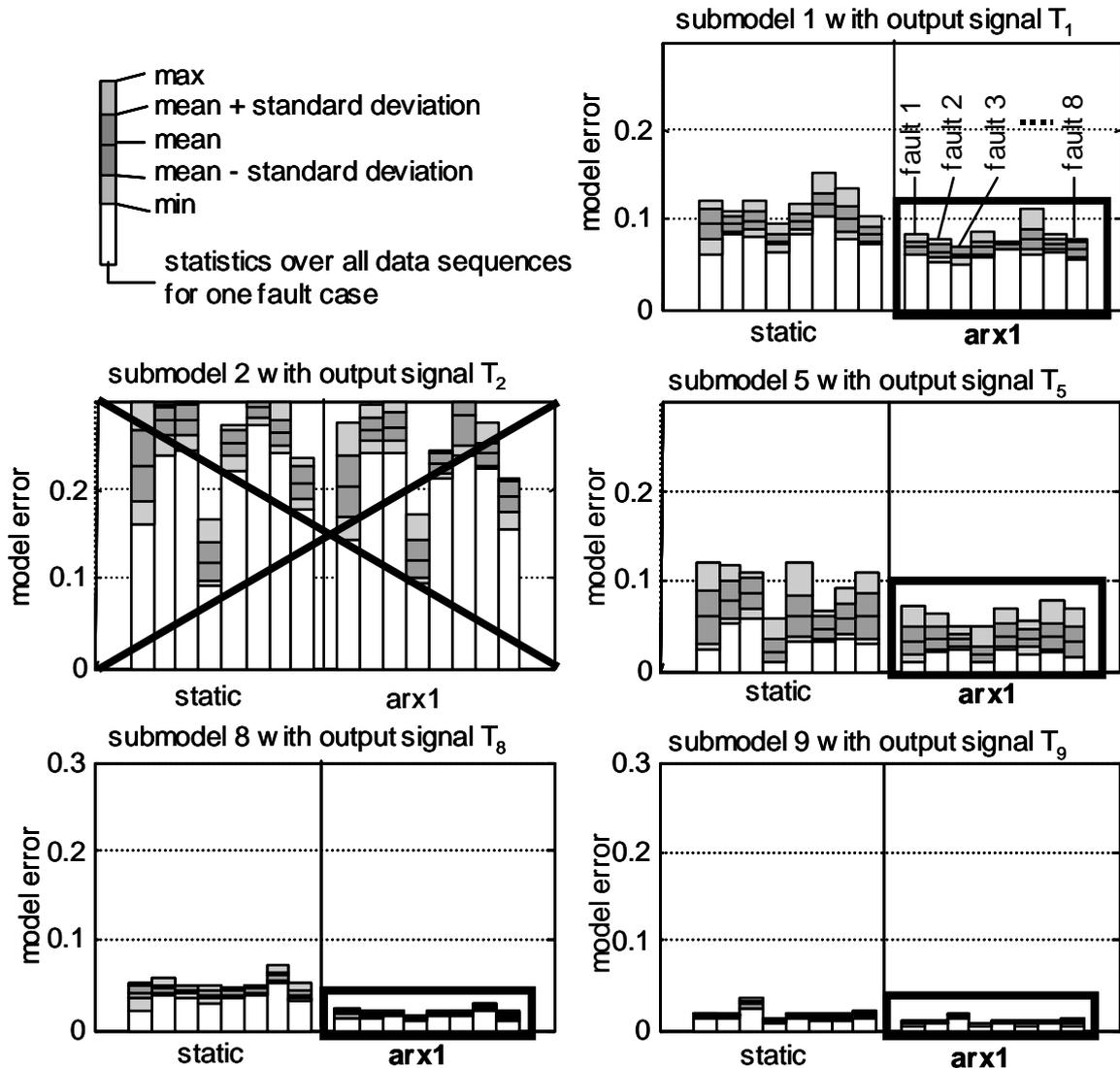


Fig. 37: Error charts for the submodels 1, 2, 5, 8, 9 with the output signals  $T_1$ ,  $T_2$ ,  $T_5$ ,  $T_8$ ,  $T_9$ , decoupled model structure.

As a result of the error charts above and the other error charts for the remaining submodels, the ARX approaches have been selected for the submodels 1 and 4..11, whereas the submodels 2 and 3 have been eliminated.

### 6.1.8 Standard Clusters

All the clustering methods of Section 5.4 have been tested for this application, such as the HCM2Fuzzy, HCM2Neuro, FCM2Neuro and PFCM2Neuro methods. Fig. 38 and Fig. 39 visualize the clusters that result from the HCM2Neuro method in Section 5.4.2, using the physical model structure. Whereas Fig. 20 on page 67 only shows the cluster for fault case 3, here all clusters for the fault cases 1..8 are shown. Remember that for these plots the 39-dimensional parameter space is projected onto two parameters, namely  $x_1=a_2$  and  $x_2=b_{2,1}$  (cf. Fig. 35). For each plot, one cluster is cut through its center, which means that the parameters  $x_1$  and  $x_2$  are varied in the range of -0.2..1.2, but the other parameters  $x_3..x_{39}$  are fixed to the coordinate of the center. For each plot, these centers are at another ‘height’ in the parameter space (other values for  $x_3..x_{39}$ ) and therefore, the seeming overlappings are much lower in the high-dimensional parameter space.

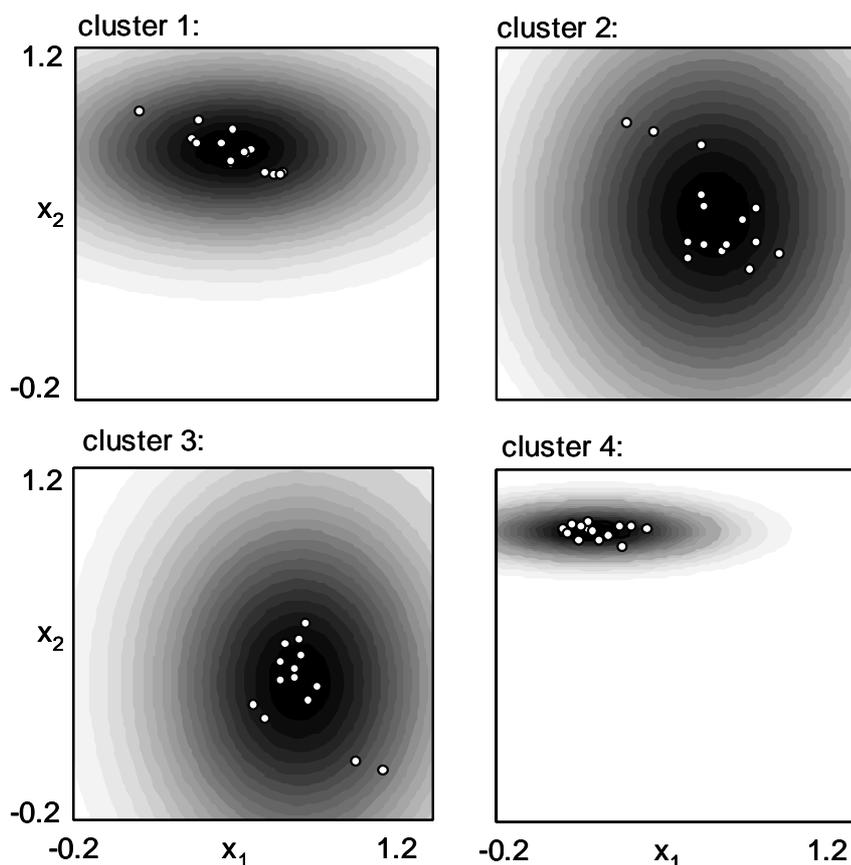


Fig. 38: Results of the HCM2Neuro method, cuts through the centers of the clusters 1..4 (faults 1..4), projection from  $n$  dimensions onto the first two parameter axes  $x_1$  and  $x_2$ . Dark = high membership grades, light = low membership grades. Data points = training parameter sets.

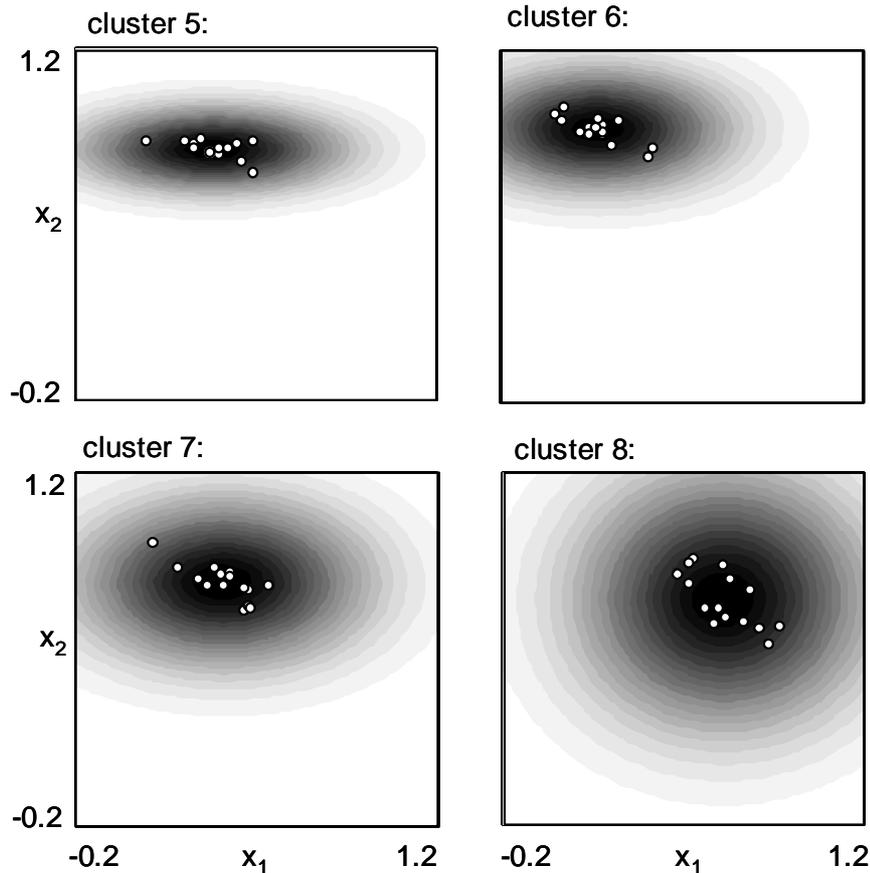


Fig. 39: Results of the HCM2Neuro method, cuts through the centers of the clusters 5..8 (faults 5..8)

These plots show that all clusters are able to enclose the training data points (= parameter sets) of the corresponding fault cases. If the data points are close to each other, then the corresponding cluster is smaller. Since the variations of the data points are different in each parameter axis direction  $x_i$ , some clusters are rather ellipsoid than spherical. The overlappings of the clusters in the high-dimensional parameter space will be tested by checking the classifications of the training data points with the algorithm of Section 5.4.4. Some outlying data points are especially observed in the clusters 2 and 3. They may cause a few additional wrong classifications in the checking algorithm and could be eliminated manually. But here all training data points are considered in order to demonstrate the generality of the method without any extra user inputs being required.

### 6.1.9 Classification Results for Standard Clusters

For checking the classification quality using the training data, the algorithm of Section 5.4.4 is applied. Of the eight faults, the separation of seven faults was possible with good results. Fault 5 affects the system similarly to fault 6, and therefore was eliminated. For 7 clusters with 15 points each, and 7 checks per point, a total of 735 classification tests are run. Table 4 shows the classification quality for the physical model structure, regarding the clustering method and the number of output signals or submodels, respectively. As a result of the model error analysis of Section 6.1.4, nine submodels 2, 4..11 were preselected with the output signals  $T_2, T_4..T_{11}$ .

*Table 4: Results for the physical model structure with different clustering methods and different sets of selected signals. Table cell contents: first line: (number of wrong classifications) / (number of checks) = rate of wrong classifications, subsequent lines: selected signals (bold = output signals, other = input signals), last line: number of required sensors.*

number of output signals	HCM-2Fuzzy clustering method	HCM-2Neuro clustering method	PFCM-2Neuro clustering method	FCM-2Neuro clustering method
9	4/735= <b>0.54%</b> <b><math>T_2, T_4..T_{11}</math></b> , $T_1, T_3, T_{12}, T_{14}$ 13 sensors	7/735= <b>0.95%</b> <b><math>T_2, T_4..T_{11}</math></b> , $T_1, T_3, T_{12}, T_{14}$ 13 sensors	18/735=2.45% <b><math>T_2, T_4..T_{11}</math></b> , $T_1, T_3, T_{12}, T_{14}$ 13 sensors	30/735=4.08% <b><math>T_2, T_4..T_{11}</math></b> , $T_1, T_3, T_{12}, T_{14}$ 13 sensors
3 of 9	7/735= <b>0.95%</b> <b><math>T_9, T_{10}, T_{11}</math></b> , $T_1, T_2, T_5..T_8,$ $T_{12}, T_{14}$ 11 sensors	7/735= <b>0.95%</b> <b><math>T_6, T_9, T_{11}</math></b> , $T_1, T_2, T_5..T_8,$ $T_{12}, T_{14}$ 11 sensors	18/735=2.45% <b><math>T_6, T_9, T_{11}</math></b> , $T_1, T_2, T_5..T_8, T_{12},$ $T_{14}$ 11 sensors	20/735=2.72% <b><math>T_5, T_9, T_{11}</math></b> , $T_1, T_2, T_4, T_7,$ $T_8, T_{12}, T_{14}$ 10 sensors
2 of 9	11/735= <b>1.50%</b> <b><math>T_9, T_{11}</math></b> , $T_1, T_2, T_7, T_8,$ $T_{12}, T_{14}$ 8 sensors	11/735= <b>1.50%</b> <b><math>T_9, T_{11}</math></b> , $T_1, T_2, T_7, T_8,$ $T_{12}, T_{14}$ 8 sensors	22/735=2.99% <b><math>T_9, T_{11}</math></b> , $T_1, T_2, T_7, T_8, T_{12},$ $T_{14}$ 8 sensors	26/735=3.54% <b><math>T_9, T_{11}</math></b> , $T_1, T_2, T_7, T_8,$ $T_{12}, T_{14}$ 8 sensors

In the first row of the table the results are presented for all nine submodels. The second and third rows present the results for a combination of three and

two submodels out of nine. For finding the submodel combinations with the best ranking (fewest wrong classifications), the method of Section 5.5.1 was used. This procedure was repeated for each cell of Table 4, where the output signals of the best submodels are printed in bold. In general, this selection varies according to the clustering method.

Table 4 shows that *HCM2Fuzzy* and *HCM2Neuro* yield the best results with a rate of wrong classifications at or below 1.5%. *PFCM2Neuro* and *FCM2Neuro* have more wrong classifications with the current data because of their shape, which is less adaptive to the data points of this application. The number of sensors required is quite high for the physical model structure because the selected output signals are coupled with a number of other signals. In the table, the coupled signals are displayed below the output signals (output signals in bold).

In order to reduce the number of measured signals, the decoupled model structure is chosen (Table 5). As a result of the model error analysis, nine submodels 1, 4..11 were preselected with the output signals  $T_1, T_4..T_{11}$ .

*Table 5: Results for the decoupled model structure with different clustering methods and different sets of selected signals. Table cell contents: first line: (number of wrong classifications) / (number of checks) = rate of wrong classifications, subsequent lines: selected signals (bold = output signals, other = input signals), last line: number of required sensors.*

number of output signals	HCM-2Fuzzy clustering method	HCM-2Neuro clustering method	PFCM-2Neuro clustering method	FCM-2Neuro clustering method
9	29/735=3.95% <b><math>T_1, T_4..T_{11}</math></b> , $T_{12}, T_{14}$ 11 sensors	27/735=3.67% <b><math>T_1, T_4..T_{11}</math></b> , $T_{12}, T_{14}$ 11 sensors	43/735=5.85% <b><math>T_1, T_4..T_{11}</math></b> , $T_{12}, T_{14}$ 11 sensors	52/735=7.07% <b><math>T_1, T_4..T_{11}</math></b> , $T_{12}, T_{14}$ 11 sensors
3 of 9	29/735= <b>3.95%</b> <b><math>T_6, T_9, T_{11}</math></b> , $T_{12}, T_{14}$ <b>5 sensors</b>	32/735= <b>4.35%</b> <b><math>T_6, T_9, T_{11}</math></b> , $T_{12}, T_{14}$ <b>5 sensors</b>	38/735=5.17% <b><math>T_5, T_9, T_{11}</math></b> , $T_{12}, T_{14}$ 5 sensors	47/735=6.39% <b><math>T_5, T_9, T_{11}</math></b> , $T_{12}, T_{14}$ 5 sensors
2 of 9	41/735= <b>5.58%</b> <b><math>T_9, T_{11}</math></b> , $T_{12}, T_{14}$ <b>4 sensors</b>	53/735= <b>7.21%</b> <b><math>T_9, T_{11}</math></b> , $T_{12}, T_{14}$ <b>4 sensors</b>	76/735=10.34% <b><math>T_8, T_9</math></b> , $T_{12}, T_{14}$ 4 sensors	88/735=11.97% <b><math>T_8, T_9</math></b> , $T_{12}, T_{14}$ 4 sensors

Here all output signals are only coupled with the two input signals  $T_{12}$  and  $T_{14}$  (and  $u_{step}$  which is not measured). Thus, the number of required sensors can be reduced to five in the case of three output signals (second row) or even to four in the case of two output signals (last row). For the *HCM2Fuzzy* clustering method, the rate of wrong classifications is below 4% with five sensors and below 6% with four sensors.

## 6.2 Data from Simulation Model (Application III)

Various fault cases are simulated by the simulation model described in Chapter 11, which is designed for an air-to-water heat pump from SATAG<sup>®</sup> Thermo-technik AG and validated with nominal data of one heating period (1998) of the residential building at Barzheim (Schaffhausen, Switzerland). The data sheet of the heat pump is given in Appendix C, Section ‘SATAG Natura AW110’. For the simulation of the fault cases, the input signals are extracted from the same data, but the model parameters are changed.

### 6.2.1 Configuration of the Simulated Heat Pump

The ‘virtual’ temperature sensors ( $T_1..T_{10}$ ) in Fig. 40 have been selected as the initial sensor configuration for the training of the fault diagnosis system. No mass flows, heat flows, or other internal signals of the simulation model are used for the fault diagnosis system, because in real applications the sensors for measuring those signals would be too expensive. Neither the power input of the compressor, nor the pressure signals are used.

Nine fault cases were simulated by changing the corresponding parameters of the simulation model (cf. Table 6). For each fault case, 21 data sequences were generated. They were separated into two different data sets, one data set for the training of the diagnosis system (containing 10 data sequences), and the other data set for its validation (containing 11 data sequences). The simulated signals were resampled at a sampling time of 10 seconds for the training of the fault diagnosis system.

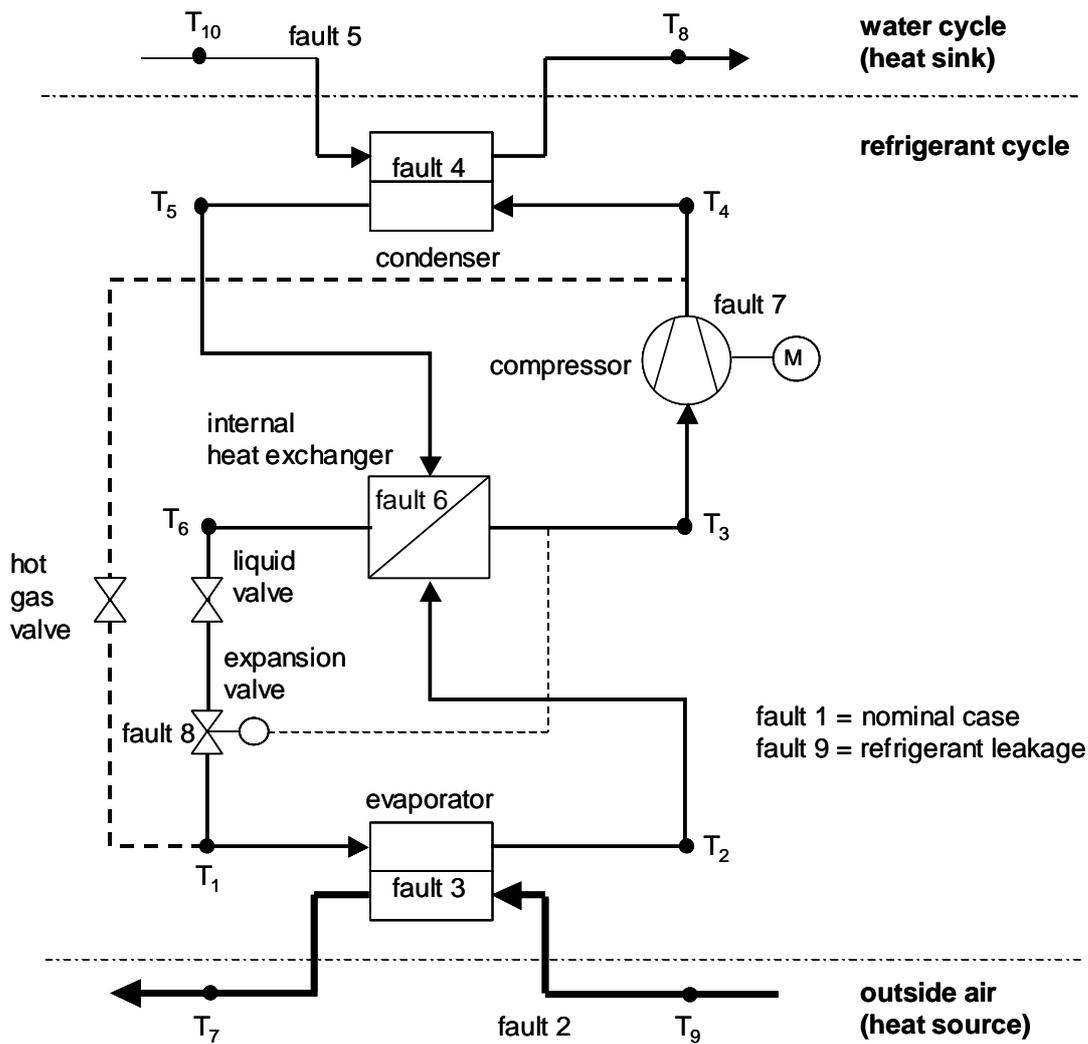


Fig. 40: Simulated air-to-water heat pump with an internal heat exchanger.

The commercial air-to-water heat pump has three modes (off, on, defrosting). Either the off-on or the defrosting-on sequences are most interesting for diagnosing faults. Since most of the simulations are made with input data from the winter period, where the defrosting-on sequences are more numerous, only those sequences are used for identifying the parameters. The duration of the data sequences was chosen to be equal to the full duration of the defrosting-on sequences in the measured data, which are in the range of 60..90 minutes.

Table 6: The simulated fault cases.

fault case	fault description (faults in real applications)
fault 1	normal behaviour, no fault
fault 2	reduced evaporator heat transfer (evaporator fouling)
fault 3	reduced evaporator air mass flow (evaporator/air channel fouling, air fan malfunction)
fault 4	reduced condenser heat transfer (condenser fouling)
fault 5	reduced condenser water mass flow (fouling, water pump malfunction)
fault 6	reduced internal heat exchanger heat transfer (internal heat exchanger fouling)
fault 7	reduced compressor efficiency (compressor malfunction)
fault 8	reduced expansion valve flow rate (expansion valve malfunction)
fault 9	reduced refrigerant charge (refrigerant leakage)

### 6.2.2 Model

The input vector  $u(t)$  and the output vector  $y(t)$  are defined in eq. (53).

$$y(t) = \begin{bmatrix} T_1(t) \\ T_2(t) \\ \vdots \\ T_8(t) \end{bmatrix} \quad u(t) = \begin{bmatrix} T_9(t) \\ T_{10}(t) \\ u_{step}(t) \end{bmatrix} \quad (53)$$

The system matrix of the physical model structure is given by eq. (54).

	$T_1$	$T_2$	$T_3$	$T_4$	$T_5$	$T_6$	$T_7$	$T_8$	$T_9$	$T_{10}$	$u_{step}$
$T_1$	×	0	0	0	0	•	0	0	0	0	•
$T_2$	0	×	0	0	0	•	•	0	•	0	•
$T_3$	0	•	×	0	•	•	0	0	0	0	•
$T_4$	0	0	•	×	0	0	0	0	0	0	•
$T_5$	0	0	0	•	×	0	0	•	0	•	•
$T_6$	0	•	•	0	•	×	0	0	0	0	•
$T_7$	0	•	0	0	0	•	×	0	•	0	•
$T_8$	0	0	0	•	•	0	0	×	0	•	•

(54)

The other model structures are in analogy to Section 5.2.

### 6.2.3 Signal Plots

For example, a nominal data sequence of the training data set is plotted in Fig. 41 and Fig. 42 (corresponding to the defrosting-on sequence starting at  $t=9.6$  minutes of January 27, 1998). All signals are scaled to a range in the region of 0..1. The measured input signals are represented by the temperatures  $T_9$ ,  $T_{10}$  and the step input  $u_{step}$ . For all submodels 1..8, the simulated output signals are compared to the measured signals  $T_1..T_8$ . Comparing to the signals of application II in Section 6.1, these output signals are more moderate. There are fewer nonlinearities in their transients, and the ‘hunting effect’ has a lower impact since the signals are simulated. Nevertheless, the disturbances act on the input signals, which are propagated in the nonlinear model. The input signals are also superposed by the noise in the measured data, which can be seen in the greatly zoomed signal  $T_9$ . The input signal  $T_{10}$  is superposed by noise of a similar variance, but here the noise is not apparent because of a much lower scaling factor.

Regarding these plots, different time constants of the output signals are obvious. The signals  $T_1$ ,  $T_2$ ,  $T_3$ , and  $T_7$  have small time constants, whereas the sig-

nal  $T_4$  has a large time constant. The signals  $T_5$ ,  $T_6$ , and  $T_8$  are characterized by a fast response at the beginning and a slow transient afterwards. The maximum time constants of the signals  $T_4$ ,  $T_5$ ,  $T_6$ , and  $T_8$  are taken into account by the duration of the data sequence, which is at about 80 minutes for the current sequence.

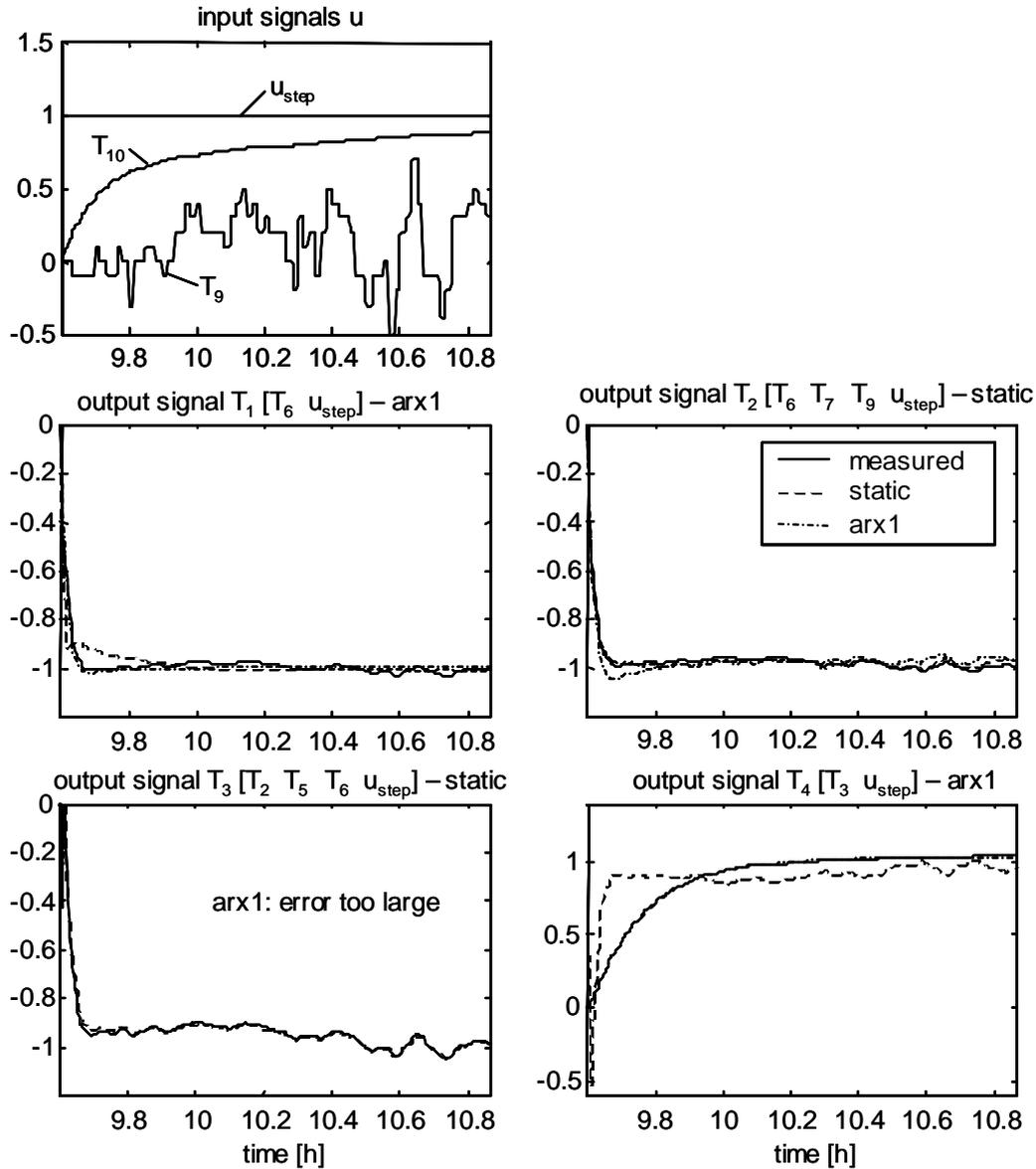


Fig. 41: Nominal data sequence, physical model structure. Upper left plot: measured input signals ( $T_9$ ,  $T_{10}$ ) and step input signal  $u_{step}$ . Other plots: measured (solid line) and simulated output signals  $T_1..T_4$  for the static and first-order ARX approaches. All signals are scaled. The coupled signals are listed in brackets [], for example:  $T_1$  is coupled with the signals  $T_6$  and  $u_{step}$ . The approaches which best fit the measured data of the current sequence are noted in the title of the plots.

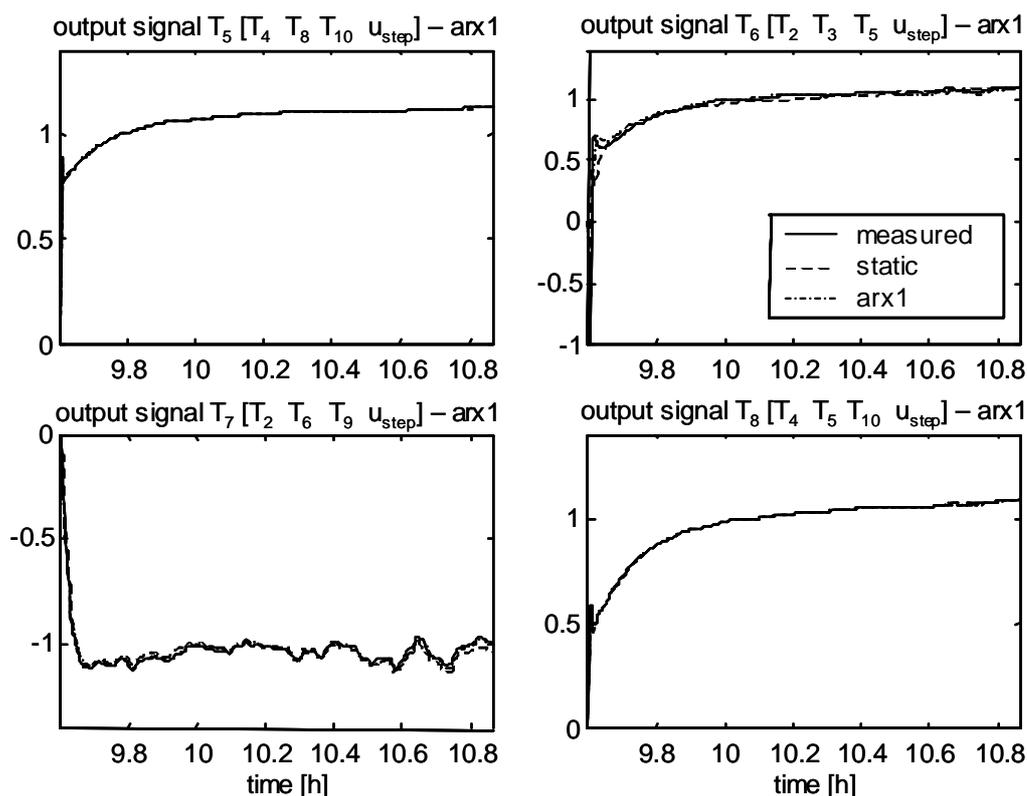


Fig. 42: Nominal data sequence, physical model structure. Measured (solid line) and simulated output signals  $T_5..T_8$ . All signals are scaled.

For each submodel of the plotted sequence, at least one of the model approaches fits the data with low model errors, such as the first-order ARX approach for the output signals  $T_1$ ,  $T_4$ , and  $T_6$ , the static approach for the output signals  $T_2$  and  $T_3$ , and both approaches for the signals  $T_5$ ,  $T_7$ , and  $T_8$ . Again, the measured signals and the fitting quality vary from one data sequence to another, and especially from one fault case to another. A statistical evaluation of the model errors will be shown in the the next section.

#### 6.2.4 Error Charts

Fig. 43 shows the statistical evaluation of the model errors over all 21 data sequences within each fault case 1..9 (1 = nominal case). The fitting quality observed from one data sequence in Fig. 41 and Fig. 42 is similar to the fitting quality of the rest of the data sequences, with a few exceptions. For example, the ARX approach of model 8 has large errors for the last fault case 9, whereas it has low errors for all other fault cases including the nominal case. Thus, it is important to analyze all fault cases before selecting the appropriate model approach. Submodel 5 has low model errors for both approaches. Thus the static model approach is selected in order to prevent over-parametrization.

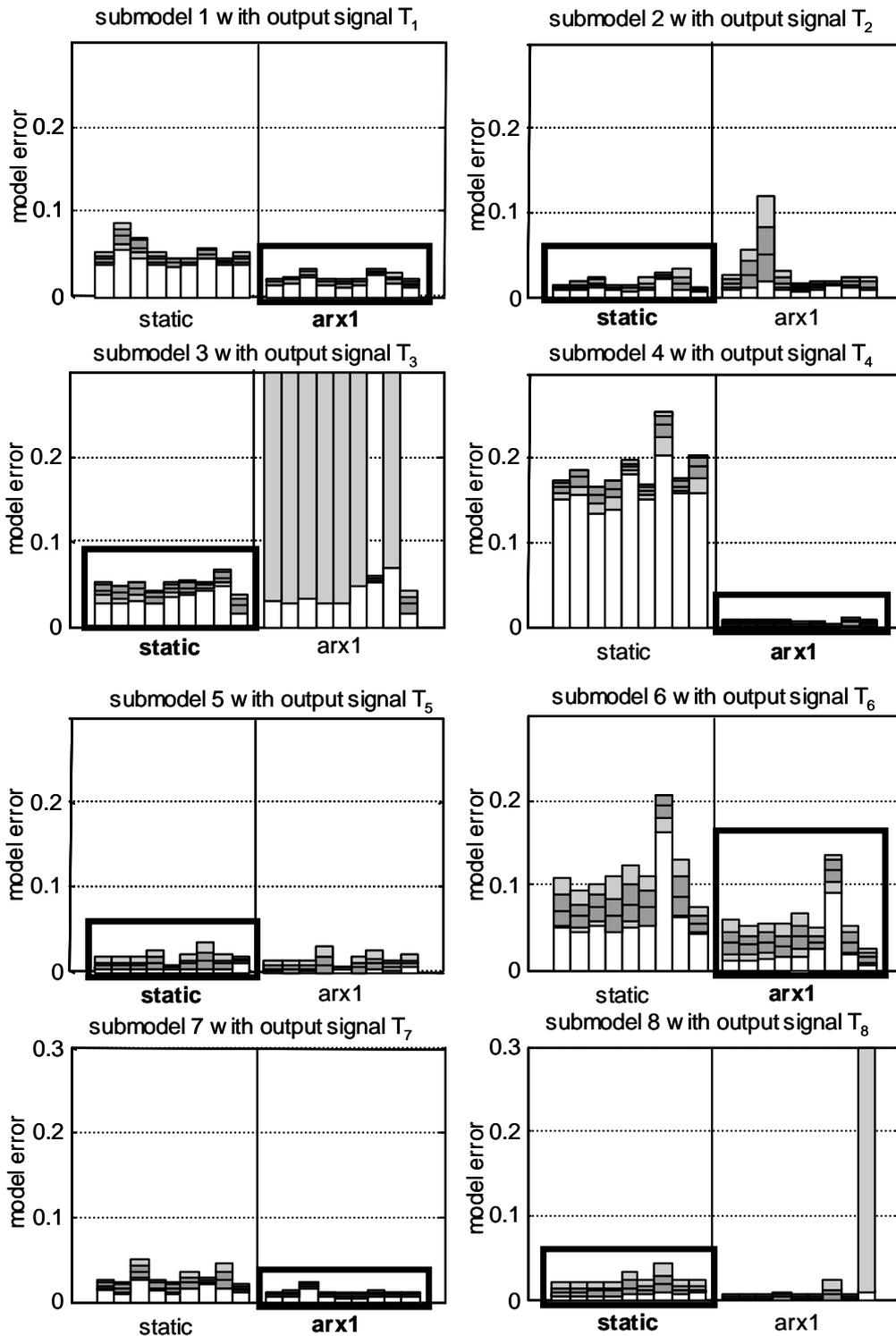


Fig. 43: Error charts for the submodels 1..8 with the output signals  $T_1..T_8$ , physical model structure. For each plot, the errors of the static model are in the left bar group, and the errors of the 1<sup>st</sup> order ARX model are in the right bar group; within a group, each bar represents one fault case (faults 1..9 from left to right). Legend for the bars: light gray region = minimum/maximum values of the errors, dark gray region = standard deviations of the errors

As a result of the error charts above, the ARX approaches were selected for the submodels 1, 4, 6, 7 and the static approaches were selected for the submodels 2, 3, 5 and 8. No submodel had to be eliminated. Together, all selected approaches have a total number of 32 parameters.

The same analysis was made for the decoupled model structure, where submodel 6 had to be eliminated from the eight submodels.

### 6.2.5 Classification Results for Standard Clusters

The standard clustering methods of Section 5.4 are used for building the clusters in the 32-dimensional parameter space, using the training data set. For all nine fault cases, the classification quality is tested by the checking algorithm of Section 5.4.4. The same algorithm is used to check the classification of the training data points (= parameter sets from the training data set) as well as the validation data points (= parameter sets from the validation data set).

Here only the *HCM2Neuro* clustering method was applied because of its higher classification quality for validation data that differs from the training data. The results are summarized in Table 7. The physical and the decoupled model structure as well as different submodel selections are investigated (rows of Table 7). The wrong classification rate obviously is higher for the validation data set (last column) than for the training data set (column before). In the first row, the results for the full physical model with eight output signals are displayed. For this case, ten sensors are required, including the eight output signals ( $T_1..T_8$ ) and the two input signals ( $T_9, T_{10}$ ). In the next row, the three best of eight submodels are selected with the output signals  $T_2, T_3, T_5$ , where the criterion is a minimum number of wrong classifications. Therefore, the algorithm of Section 5.5.1 is used. Because of the couplings in the physical model, the output signals  $T_2, T_3, T_5$  depend on the other signals  $T_4$  and  $T_6..T_{10}$ , which adds up to nine sensors. Selecting two of eight submodels with the output signals  $T_2$  and  $T_3$ , the number of sensors is reduced to six. For a further sensor reduction, the model has to be switched to the decoupled structure, where a total of seven submodels are preselected (without submodel 6). Here the number of sensors could be reduced to five or even four. But the rate of wrong classifications is too high for the validation data set (last row). Therefore, as a compromise between a minimal number of sensors and a minimal wrong classification rate, the physical model with six sensors (third row) is chosen.

Table 7: Results for the physical and decoupled model structures with the HCM2Neuro clustering method, different sets of selected signals (bold = output signals). Table cell contents: (number of wrong classifications) / (number of checks) = rate of wrong classifications

Model, number of output signals	signals required, number of sensors	training data wrong class. rate	validation data wrong class. rate
physical 8 output signals	<b><math>T_1..T_8</math></b> , $T_9$ , $T_{10}$ 10 sensors	11/720= <b>1.5%</b>	24/792= <b>3.0%</b>
physical 3 of 8 output sig.	<b><math>T_2</math>, <math>T_3</math>, <math>T_5</math></b> , $T_4$ , $T_6..T_{10}$ 9 sensors	23/720= <b>3.2%</b>	72/792= <b>9.1%</b>
<b>physical 2 of 8 output sig.</b>	<b><math>T_2</math>, <math>T_3</math>, <math>T_5</math></b> , $T_6$ , $T_7$ , $T_9$ 6 sensors	43/720= <b>6.0%</b>	70/792= <b>8.8%</b>
decoupled 3 of 7 output sig.	<b><math>T_1</math>, <math>T_4</math>, <math>T_7</math></b> , $T_9$ , $T_{10}$ 5 sensors	45/720=6.3%	164/792=20.7%

### 6.2.6 Classification Results for Vector Clusters

Eight fault vectors are defined, with the nominal case (= fault 1) as their origin and the fault cases 2..9 as their directions. In order to check the classification of gradual faults, data sets for different fault sizes were generated by the simulation model. For each fault size, the corresponding parameters of the model were changed (cf. Section 11.2.3). For the faults 2..8, four sizes were simulated:

- fault size 2: 20% parameter change
- fault size 3: 30% parameter change
- fault size 4: 40% parameter change

For fault 9, the three simulated sizes are smaller, since its effect on the signals is larger:

- fault size 1: 10% parameter change
- fault size 2: 20% parameter change
- fault size 3: 30% parameter change

The classification system has been trained by data for the faults of size 3, and checked by validation data for the faults of the sizes 1, 2, and 4. For fault case 9, size 2 was trained and the sizes 1 and 3 were checked.

In Table 8, the membership grades  $f_i(x)$  to the fault vectors  $i$  are shown (cf. eq. (45) on page 80). The fault vectors  $i$  are represented by the columns of the table, and the data sets for the fault cases with the increasing fault sizes are represented by the rows. Here, the membership grades are not shown for each single data point, but they are averaged over the full data sets, which contain 21 data points per fault size.

Regarding the rows of Table 8, the cells with maximal membership grades are displayed with frames. The cells of the corresponding fault vectors are displayed in gray. If the locations of the gray cells and the framed cells are equal, then the classification is correct. This is always the case, with one exception only (data set for fault case 9, size 1). The membership grades close to one indicate that the data points of the corresponding fault cases are close to their fault vectors. For fault case 7, the fault sizes 2, 3, and 4 lie exactly on the fault vector. Thus the assumption of a linear fault trajectory is fulfilled for this fault case. For the other fault cases, the variations are larger, but they are also classified correctly (with one exception).

Regarding all fault sizes of fault case 4, some rather high membership grades are observed for fault vector 9, and vice versa for all fault sizes of fault case 9 some rather high membership grades are observed for fault vector 4. This means that the fault vectors 4 and 9 are close to each other. Some physical explanation can be found by remembering, that fault case 4 corresponds to a reduced condenser heat transfer and fault case 9 corresponds to refrigerant leakage. Due to refrigerant leakage, the liquid level in the condenser will be lower, which reduces the condenser heat transfer. Similar observations are made for the fault cases 6 and 8, regarding the fault vectors 8 and 6. Thus a reduction of the mass flow through the expansion valve (fault 8) is related to a reduction of the heat transfer in the internal heat exchanger (fault 6), which can be explained by the lower filling grade of the internal heat exchanger. Aside from these physical couplings of the fault cases, the vector clusters are able to classify the faults with a high quality, even for different fault sizes.

Table 8: Mean membership grades for gradual faults. Columns = fault vectors 2..9, rows = data sets for different fault cases and fault sizes (fault 1 = nominal case, faults of size 3 are trained), in the gray cells should be the maximal membership grades, in the framed cells really are the maximal membership grades

	vec. 2	vec. 3	vec. 4	vec. 5	vec. 6	vec. 7	vec. 8	vec. 9
fault 1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
fault 2, size 1	0.758	0.000	0.180	0.000	0.000	0.000	0.000	0.000
fault 2, size 2	0.894	0.000	0.135	0.000	0.000	0.000	0.000	0.000
fault 2, size 3	0.946	0.000	0.105	0.000	0.000	0.000	0.000	0.000
fault 2, size 4	1.000	0.000	0.052	0.000	0.000	0.046	0.000	0.000
fault 3, size 1	0.000	0.742	0.000	0.000	0.000	0.000	0.000	0.000
fault 3, size 2	0.000	0.865	0.091	0.000	0.000	0.000	0.000	0.000
fault 3, size 3	0.000	0.925	0.072	0.000	0.000	0.000	0.000	0.000
fault 3, size 4	0.000	0.952	0.048	0.000	0.000	0.000	0.000	0.000
fault 4, size 1	0.000	0.000	0.392	0.000	0.000	0.000	0.000	0.233
fault 4, size 2	0.000	0.000	0.726	0.000	0.000	0.000	0.000	0.438
fault 4, size 3	0.106	0.000	0.862	0.000	0.000	0.000	0.000	0.543
fault 4, size 4	0.079	0.000	0.904	0.000	0.000	0.000	0.000	0.595
fault 5, size 1	0.000	0.000	0.000	0.728	0.000	0.000	0.000	0.000
fault 5, size 2	0.000	0.000	0.000	0.915	0.204	0.000	0.000	0.146
fault 5, size 3	0.000	0.000	0.000	0.979	0.157	0.000	0.000	0.091
fault 5, size 4	0.000	0.000	0.000	0.991	0.193	0.095	0.000	0.000
fault 6, size 1	0.000	0.000	0.000	0.000	0.640	0.000	0.325	0.000
fault 6, size 2	0.000	0.000	0.000	0.162	0.807	0.000	0.333	0.000
fault 6, size 3	0.000	0.000	0.000	0.088	0.978	0.000	0.141	0.000
fault 6, size 4	0.000	0.000	0.000	0.081	1.000	0.000	0.064	0.000
fault 7, size 1	0.000	0.000	0.000	0.839	0.237	0.940	0.000	0.000
fault 7, size 2	0.000	0.000	0.000	0.473	0.109	1.000	0.000	0.000
fault 7, size 3	0.028	0.000	0.000	0.040	0.005	1.000	0.000	0.000
fault 7, size 4	0.044	0.004	0.016	0.027	0.000	1.000	0.000	0.000
fault 8, size 1	0.000	0.000	0.000	0.000	0.000	0.000	0.775	0.000
fault 8, size 2	0.000	0.000	0.000	0.000	0.194	0.000	0.926	0.000
fault 8, size 3	0.000	0.013	0.000	0.000	0.199	0.000	0.989	0.000
fault 8, size 4	0.000	0.042	0.000	0.000	0.241	0.082	0.997	0.000
fault 9, size 1	0.097	0.000	0.797	0.000	0.000	0.000	0.000	0.737
fault 9, size 2	0.045	0.000	0.636	0.068	0.000	0.000	0.000	0.926
fault 9, size 3	0.000	0.000	0.348	0.438	0.000	0.000	0.000	0.688

The fault grades  $g_i(x)$  (eq. (43) on page 79) are also averaged and shown in Table 9. According to eq. (50) on page 84, a minimum fault grade  $g_{min}$  is introduced, which is set to 0.1. If the fault grades  $g_i(x)$  are lower than  $g_{min}$ , then the corresponding membership grades  $f_i(x)$  in Table 8 are set to zero. The mean fault grades of the nominal data (fault 1) are obviously zero (Table 9, first row). For the trained faults of size 3 (or size 2 for fault case 9), the mean fault grades are one, because the fault grades are scaled by the trained values. With data of an increasing fault size (1 to 4), the diagnosed fault

grades increase as well (all cells of a gray block). There is only one exception to that for fault case 9, size 3. Regarding the membership grades of Table 8, fault 9 moves away from the linear vector for fault sizes larger than two.

Table 9: Mean fault grades for gradual faults. Columns = fault vectors, rows = data sets for different fault cases and fault sizes (fault 1 = nominal case, faults of size 3 are trained), in the gray cells are the increasing fault grades corresponding to the increasing fault sizes

	vec. 2	vec. 3	vec. 4	vec. 5	vec. 6	vec. 7	vec. 8	vec. 9
fault 1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
fault 2, size 1	0.396	0.059	0.178	-0.090	-0.178	0.025	-0.100	0.028
fault 2, size 2	0.661	0.061	0.283	-0.140	-0.285	0.046	-0.168	0.045
fault 2, size 3	1.000	0.019	0.397	-0.186	-0.407	0.076	-0.262	0.066
fault 2, size 4	1.454	-0.094	0.523	-0.226	-0.552	0.121	-0.392	0.090
fault 3, size 1	-0.042	0.492	0.073	-0.115	-0.036	-0.008	0.063	-0.007
fault 3, size 2	-0.027	0.745	0.125	-0.166	-0.084	-0.005	0.072	-0.008
fault 3, size 3	0.016	1.000	0.189	-0.210	-0.155	0.007	0.063	-0.008
fault 3, size 4	0.088	1.251	0.262	-0.241	-0.251	0.030	0.033	-0.007
fault 4, size 1	0.060	0.039	0.412	-0.055	-0.170	-0.004	-0.088	0.140
fault 4, size 2	0.095	0.059	0.680	-0.076	-0.268	-0.004	-0.154	0.235
fault 4, size 3	0.134	0.078	1.000	-0.089	-0.378	-0.001	-0.239	0.351
fault 4, size 4	0.175	0.097	1.375	-0.083	-0.490	0.008	-0.349	0.490
fault 5, size 1	-0.079	-0.097	-0.032	0.462	0.064	0.037	-0.103	0.087
fault 5, size 2	-0.135	-0.167	-0.067	0.748	0.115	0.058	-0.157	0.136
fault 5, size 3	-0.183	-0.252	-0.257	1.000	0.222	0.090	-0.177	0.122
fault 5, size 4	-0.213	-0.370	-0.672	1.196	0.398	0.141	-0.139	0.015
fault 6, size 1	-0.169	-0.116	-0.493	0.045	0.301	-0.016	0.173	-0.175
fault 6, size 2	-0.301	-0.176	-0.847	0.114	0.571	-0.010	0.278	-0.295
fault 6, size 3	-0.485	-0.226	-1.331	0.270	1.000	0.025	0.379	-0.457
fault 6, size 4	-0.666	-0.222	-1.888	0.501	1.525	0.098	0.433	-0.656
fault 7, size 1	-0.075	-0.289	-0.559	0.927	0.411	0.354	-0.254	-0.174
fault 7, size 2	0.034	-0.373	-0.739	1.194	0.519	0.622	-0.279	-0.273
fault 7, size 3	0.568	0.069	-0.026	0.693	0.160	1.000	0.062	-0.239
fault 7, size 4	0.747	0.234	0.632	0.567	-0.215	1.090	-0.106	-0.099
fault 8, size 1	-0.074	0.022	-0.238	-0.200	0.013	-0.037	0.377	-0.091
fault 8, size 2	-0.176	0.047	-0.484	-0.267	0.135	-0.038	0.651	-0.173
fault 8, size 3	-0.343	0.101	-0.925	-0.236	0.416	0.011	1.000	-0.311
fault 8, size 4	-0.507	0.229	-1.542	-0.112	0.744	0.138	1.441	-0.509
fault 9, size 1	0.104	0.063	0.827	-0.067	-0.363	-0.031	-0.136	0.337
fault 9, size 2	0.135	-0.020	2.124	0.255	-0.786	-0.065	-0.487	1.000
fault 9, size 3	-0.096	-0.285	1.387	1.115	-0.492	-0.051	-0.425	0.963

Apart from very few exceptions, these results show that the vector clusters are able to classify gradual faults with a high quality.

### 6.3 Data from IMRT Test Bench (Application IV)

The diagnosis system of Chapter 5 has also been tested with the data of the IMRT test bench. The software tool of Chapter 7 was used to train the diagnosis system and to test the online fault diagnosis. In contrast to the previous sections, here the averaging method of Section 5.7.2 is used, which significantly improves the classification results for online diagnosis. For comparison, the training and validation results without averaging are also presented.

#### 6.3.1 Configuration of the Test Heat Pump

The commercial brine-to-water heat pump of the IMRT test bench is modified by adding actuators and sensors for introducing faults and measuring their effects on the signals (Fig. 44).

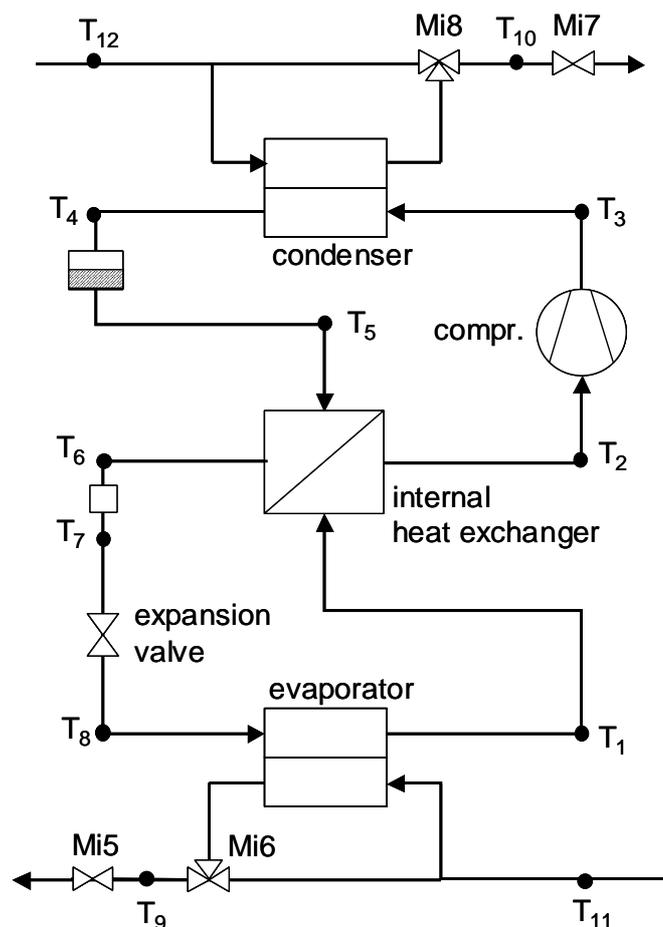


Fig. 44: The modified brine-to-water heat pump, with additional valves (Mi5...Mi8) and temperature sensors ( $T_1..T_{12}$ ).

A data sheet of the heat pump from SATAG Thermotechnik AG is shown in Appendix C, Section ‘SATAG Natura BW108’ while the test bench is described in Chapter 12. Here, the temperature sensors shown in Fig. 44 are used in the initial sensor configuration. Five fault cases are investigated, which are listed in Table 10.

Table 10: Faults of a brine-to-water heat pump.

fault	actuator	fault description
fault 1	-	normal behaviour, no fault
fault 2	valve Mi5	reduced brine mass flow (brine pump malfunction, brine cycle fouling)
fault 3	3-way valve Mi6	reduced evaporator heat transfer (evaporator fouling)
fault 4	valve Mi7	reduced water mass flow (water pump malfunction, water cycle fouling)
fault 5	3-way valve Mi8	reduced condenser heat transfer (condenser fouling)

For each fault case, data sets of four different fault sizes are measured, such as the sizes of 10%, 20%, 30%, and 50%. A size of 100% is defined as the limiting size where the heat pump is shut down because of a high-pressure or low-pressure alarm or where the ‘hunting effect’ of the expansion valve becomes significantly larger. For the size of 20%, about 40 data sequences were measured per fault case (varying for each fault case), which were then separated into a training data set and a validation data set with about 20 data sequences each. For the other sizes of 10%, 30%, and 50%, about 10..20 data sequences were measured per fault case (cf. Sections 6.2.5 and 6.2.6). The sampling time was chosen to be 1 second and the data sequences were cut to an equal length of 10 minutes.

### 6.3.2 Model

The input vector  $u(t)$  and the output vector  $y(t)$  are defined in eq. (55).

$$y(t) = \begin{bmatrix} T_1(t) \\ T_2(t) \\ \vdots \\ T_{10}(t) \end{bmatrix} \quad u(t) = \begin{bmatrix} T_{11}(t) \\ T_{12}(t) \\ u_{step}(t) \end{bmatrix} \quad (55)$$

Here the full model structure is used, in which the output signals are coupled to all other signals (cf. Section 5.2.3). The advantages of the full structure are its flexibility and high accuracy. The reduction algorithm of Section 5.5.2 is applied in order to select a subset of the output signals, which is straightforward for the full structure. The number of selected output signals and thus the number of required sensors were successively increased, starting from two output signals and four sensors.

**Two output signals.** In the first step, only those sensors are taken into account, which are already available in the commercial heat pump. Therefore, the two output signals  $T_9$ ,  $T_{10}$  (brine and water output temperatures) are selected. Together with the two input signals  $T_{11}$ ,  $T_{12}$  (brine and water input temperatures), four signals are measured.

**Three output signals.** In the next step, one sensor is added. The best classification results are obtained with the additional output signal  $T_4$  (refrigerant temperature after the condenser). With the three output signals  $T_4$ ,  $T_9$ ,  $T_{10}$  and the two input signals  $T_{11}$ ,  $T_{12}$ , five sensors are required.

Since the results are found to be satisfactory (cf. next sections), no additional sensors have to be added.

### 6.3.3 Signal Plots for Two Output Signals

For example, an off-on data sequence of the nominal case (= fault case 1) is shown in Fig. 45 for the case of two output signals. Four sensors measure the brine and water input temperatures  $T_{11}$  and  $T_{12}$  as well as the brine and water output temperatures  $T_9$  and  $T_{10}$ . The output signals are coupled to all other signals, which results in a high fitting quality. For this sequence, the model errors are low for both submodels 9 and 10 and for both model approaches. The signals contain a rather fast transient part at the beginning and a ‘drifting’ part afterwards, due to the large time constant of the test environment (representing the building).

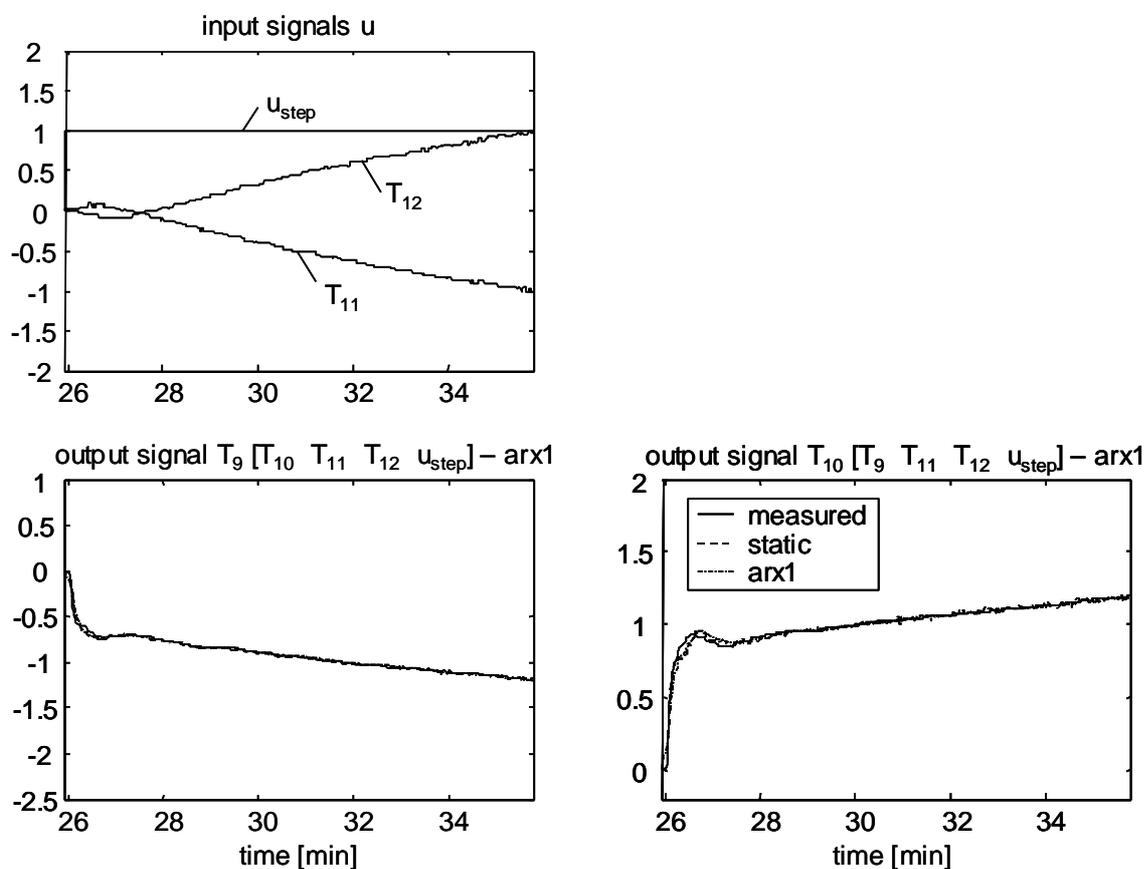


Fig. 45: Nominal data sequence, full model structure for two output signals. Upper left plot: measured input signals ( $T_{11}$ ,  $T_{12}$ ) and step input signal  $u_{step}$ . Other plots: measured (solid line) and simulated output signals  $T_9$ ,  $T_{10}$  for the static and first-order ARX approaches. All signals are scaled. The coupled signals are listed in brackets [], for example:  $T_9$  is coupled with the signals  $T_{10}$ ,  $T_{11}$ ,  $T_{12}$ , and  $u_{step}$ . The approaches which best fit the measured data of the current sequence are noted in the title of the plots.

Fig. 46 presents a data sequence of fault case 2, which is a reduced brine mass flow. As a result of that fault, the brine output temperature  $T_9$  is at a lower level. Here, the model errors are larger than for the nominal case, especially with the first-order ARX approach for submodel 10 with the output signal  $T_{10}$ . For submodel 9, the static approach shows a larger error than the ARX approach. Evidently, the model errors vary from one data sequence to another.

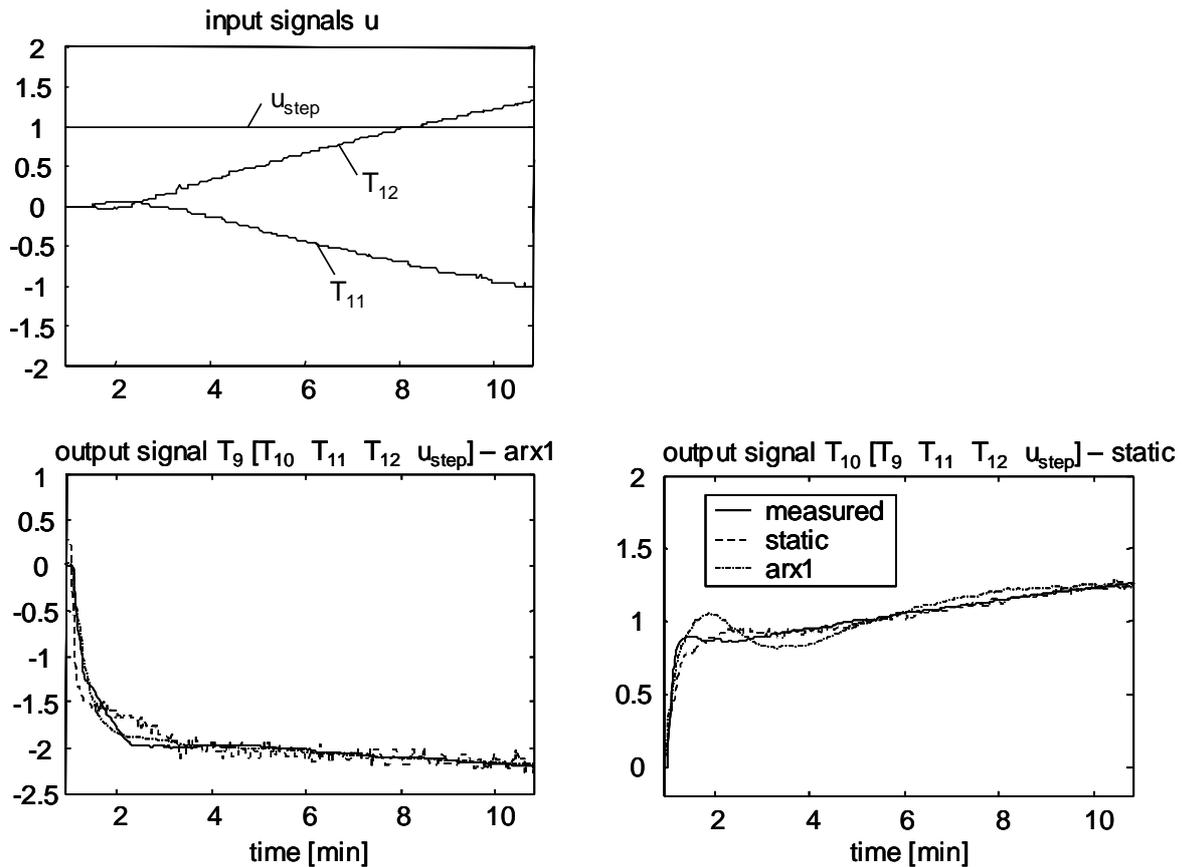


Fig. 46: Data sequence of fault 2 (reduced brine mass flow), fault size 30%, full model structure for two output signals. Upper left plot: measured input signals ( $T_{11}$ ,  $T_{12}$ ) and step input signal  $u_{step}$ . Other plots: measured (solid line) and simulated output signals  $T_9$ ,  $T_{10}$  for the static and first-order ARX approaches. All signals are scaled.

Fig. 47 presents a data sequence of fault case 3, which is a reduced evaporator heat transfer. As a result, the brine output  $T_9$  is at a higher level than in the nominal case (Fig. 45). Here, the output signal  $T_{10}$  is fitted better by the ARX approach, which is not the case for other data sequences of the same fault case.

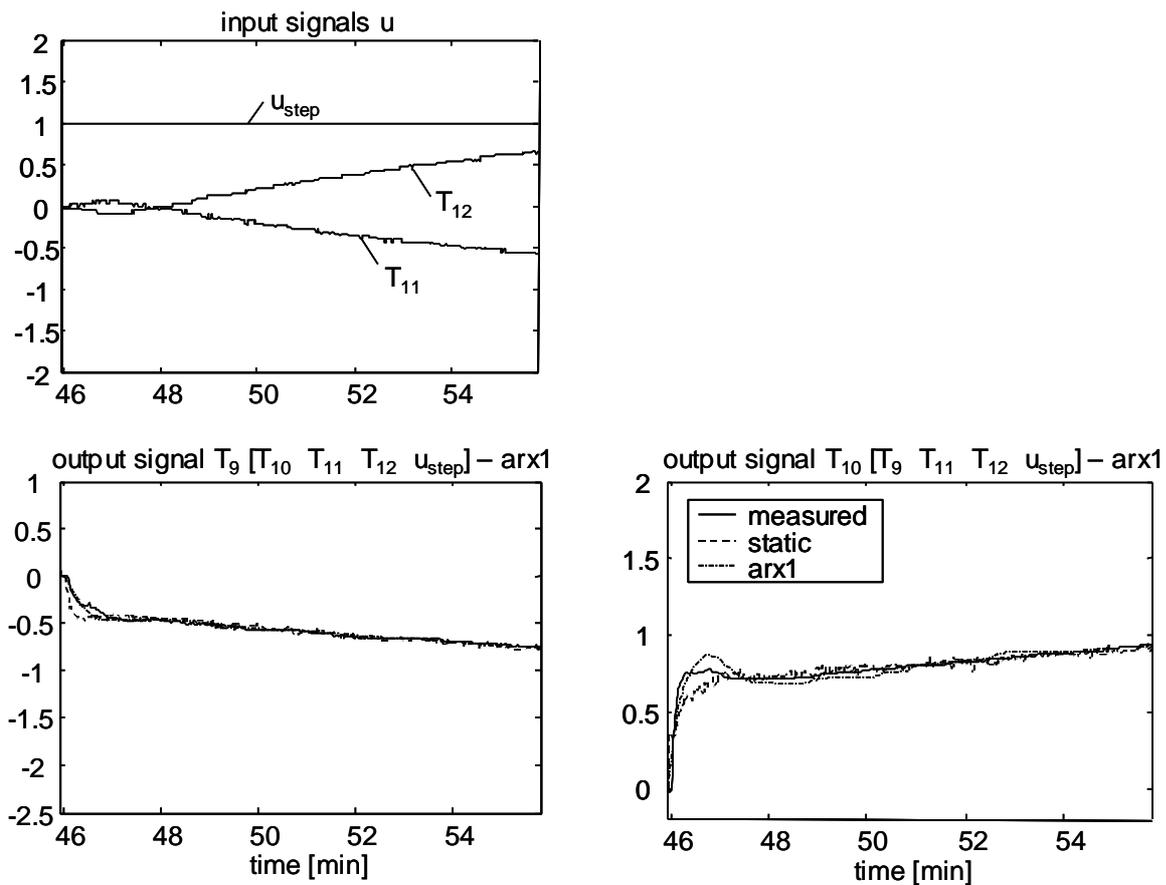


Fig. 47: Data sequence of fault 3 (reduced evaporator heat transfer), fault size 30%, full model structure for two output signals. Upper left plot: measured input signals ( $T_{11}$ ,  $T_{12}$ ) and step input signal  $u_{step}$ . Other plots: measured (solid line) and simulated output signals  $T_9$ ,  $T_{10}$  for the static and first-order ARX approaches. All signals are scaled.

### 6.3.4 Error Charts and Parameter Charts for Two Output Signals

As shown in the previous section, the model errors may change significantly from one fault case to another or even from one data sequence to another. Therefore, the model errors are statistically evaluated as shown in Fig. 48. Regarding submodel 9, the ARX approaches are better suited for the fault cases 1..3, but their errors are much higher for other fault cases 4..5. Regarding submodel 10, the model errors of the ARX approach are too high for fault case 2. Thus, the static model approaches are preselected for both submodels.

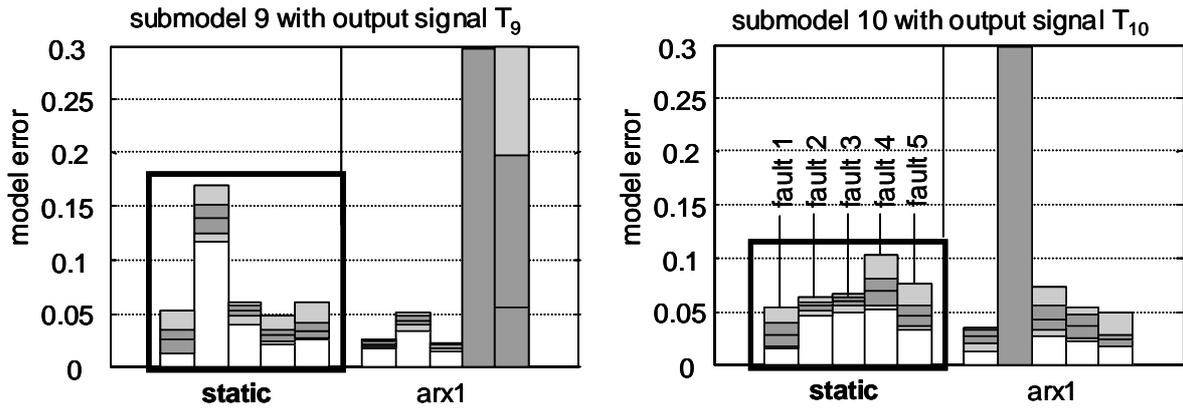


Fig. 48: Error charts for the submodels 9 and 10 with the output signals  $T_9$  and  $T_{10}$ , full model structure for two output signals. For each plot, the errors of the static model are in the left bar group, and the errors of the 1<sup>st</sup> order ARX model are in the right bar group; within a group, each bar represents one fault case (faults 1..5 from left to right). Legend for the bars: light gray region = minimum/maximum values of the errors, dark gray region = standard deviations of the errors

The parameter charts of the preselected submodels with the static approaches are shown in Fig. 49. Apparently, the separation of the faults is possible with a high classification quality. Especially for the parameters  $k_{9,10}$  and  $k_{10,9}$  the sensitivities to the faults are very high and the variances are low. For these two submodels, the parameter space will have a dimension of 8.

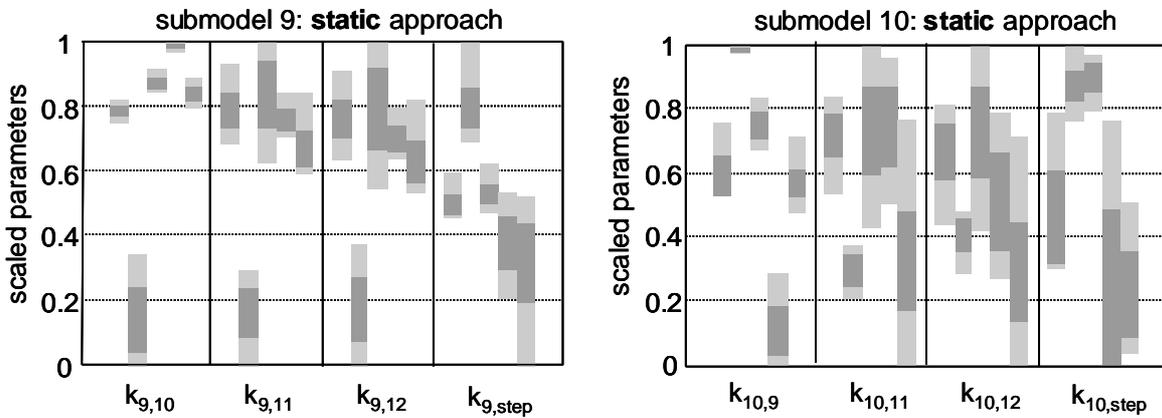


Fig. 49: Parameter charts for the submodels 9 and 10 with the static approaches selected. Within a plot, each bar group represents one parameter (plot for submodel 9: parameters  $k_{9,10}$ ,  $k_{9,11}$ , ... from left to right); within a group, each bar represents one fault case (faults 1..5 from left to right). Legend for the bars: light gray region = minimum/maximum parameter values, dark gray region = standard deviations of the parameters. All parameters are scaled.

### 6.3.5 Signal Plots for Three Output Signals

For comparison, the same data sequence as shown in Fig. 45 is plotted for three output signals in Fig. 50. The output signals  $T_4$ ,  $T_9$ , and  $T_{10}$  are coupled to all other signals, which adds up to five coupled signals per output signal. Together with the input signals  $T_{11}$  and  $T_{12}$ , a total of five signals are measured. Thus, the fitting quality is even higher for the output signals  $T_9$  and  $T_{10}$ . The model errors are also low for the additional output signal  $T_4$ . Evidently, the model errors also vary for different data sequences or even different fault cases.

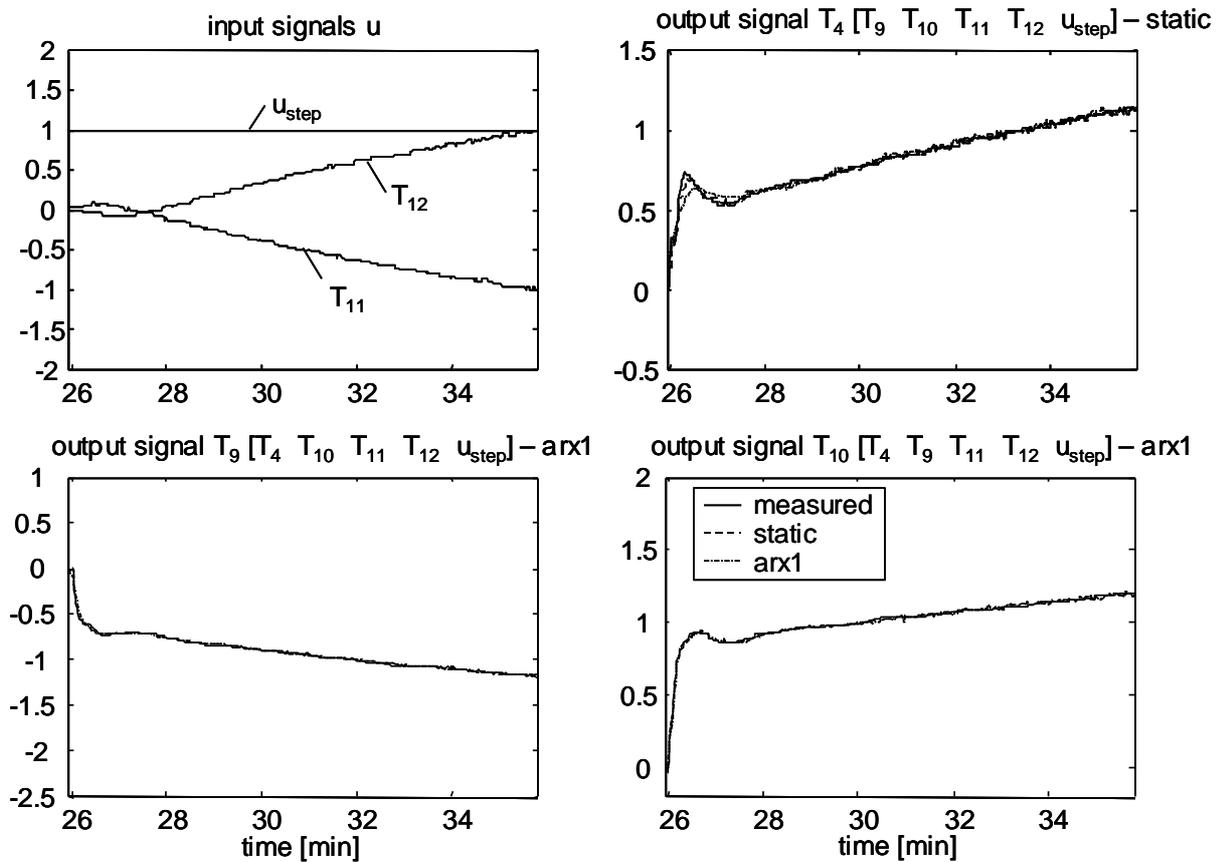


Fig. 50: Nominal data sequence, full model structure for three output signals. Upper left plot: measured input signals ( $T_{11}$ ,  $T_{12}$ ) and step input signal  $u_{step}$ . Other plots: measured (solid line) and simulated output signals  $T_4$ ,  $T_9$ ,  $T_{10}$  for the static and first order ARX approaches. All signals are scaled.

Here, the static model approaches will be selected also, since their model errors are low for all data sequences of all fault cases. Thus, the parameter space will have a dimension of 15 for these three submodels.

### 6.3.6 Classification Results for Standard Clusters

First, the *HCM2Neuro* clustering algorithm of Section 5.4.2 was used to build the clusters in the parameter space. For two output signals (four sensors) the dimension of the parameter space is 8, and for three output signals (five sensors), the dimension is 15. The resulting classification systems were tested with the training data set as well as with the validation data set (Table 11, first two columns). These data sets contain all fault cases for the size of 20%. Additionally, the same validation data set and additional data sets for the sizes 10%, 30%, and 50% were used for tests in combination with the averaging method of Section 5.7.2 (Table 11, third column). That method is developed for online diagnosis, but it is used here offline for the data sets which first were acquired and stored in a PC.

Note that the standard clusters, such as the *HCM2Neuro* clusters, are primarily designed to classify faults of one size only. Here, the faults of size 20% are trained using the training data set. No other fault sizes are trained in order to keep the experimental effort low. Therefore, each cluster corresponds to one fault case of the size 20%. The first column of Table 11 shows the rate of wrong classifications for the case of two output signals (four sensors) and the case of three output signals (five sensors), using the training data set for checking. For both cases, the classification quality is similar. In the second column, the results are given for the validation data set. As expected, the rate of wrong classifications differs from the training data set, and it is increased in the case of five sensors.

But here, the focus is on the difference between the validation data sets without and with averaging. In the last column of Table 11, the number of wrong classifications is further reduced by averaging, which results in zero wrong classifications for the faults of size 20%. For each cycle, the parameters of five data sequences are averaged before classification (cf. Section 5.7.2). Additionally, the classification quality is still high for the fault sizes 10% and 30%, even if the clusters are only trained for the fault size 20%. Thus, averaging significantly increases the robustness of the classification system. Evidently, an additional training of the fault sizes 10%, 30%, and 50% would result in even higher classification qualities for these sizes, but then the experimental effort for the training would be four times higher.

Table 11: Rate of wrong classifications for HCM2Neuro clusters with gradual faults. The faults of size 20% are used for training, the faults of the sizes 10%, 20%, 30%, and 50% are used for validation. The number of sensors is either 4 or 5.

fault size	training data set, offline, no averaging		validation data set, offline, no averaging		validation data set, "online", averaging over 5 seq./cycle	
	4 sensors	5 sensors	4 sensors	5 sensors	4 sensors	5 sensors
10%					9.7%	0.0%
20%	<b>8.3%</b>	<b>9.1%</b>	<b>6.5%</b>	<b>9.7%</b>	<b>0.0%</b>	<b>0.0%</b>
30%					3.3%	6.6%
50%					10.5%	10.5%

### 6.3.7 Classification Results for Vector Clusters

In the next step, vector clusters are built, which are especially qualified for classifying faults of different sizes. Here also, only the faults of size 20% are trained by the same training data set as in the previous section. Afterwards, the vector clusters are tested with the validation data sets (Table 12). Here, the use of averaging results in zero wrong classifications, not only for the faults of size 20%, but also for the faults of size 30%. For the other fault sizes, the results are generally better than the results of Table 11, since the vector clusters are especially qualified for gradual faults. Moreover, the sizes of the faults are calculated by the fault grades as well.

Table 12: Rate of wrong classifications for vector clusters with gradual faults. The faults of size 20% are used for training, the faults of the sizes 10%, 20%, 30%, and 50% are used for validation. The number of sensors is either 4 or 5.

fault size	training data set, offline, no averaging		validation data set, offline, no averaging		validation data set, "online", averaging over 5 seq./cycle	
	4 sensors	5 sensors	4 sensors	5 sensors	4 sensors	5 sensors
10%					6.5%	0.0%
20%	<b>4.8%</b>	<b>4.8%</b>	<b>1.2%</b>	<b>3.5%</b>	<b>0.0%</b>	<b>0.0%</b>
30%					0.0%	0.0%
50%					5.3%	5.3%

### 6.3.8 False Alarm Tuning

For all online classification systems of the previous sections, the rate of false alarm has been tuned to zero. For the HCM2Neuro clusters, the nominal weighting factor  $w_{nom}$  has to be adjusted, whereas for the vector clusters, the

minimal fault grade  $g_{min}$  has to be adjusted (cf. Section 5.7.4). For the actual data, a nominal weighting factor of 1 is sufficient, because the averaging already reduces the false-alarm rate to zero. A weighting factor higher than 1 would increase the “false-alarm robustness”. The minimal fault grade is selected for each classification system, such that no wrong classifications occur with the actual data.

Table 13: False-alarm rate for HCM2Neuro clusters and vector clusters with gradual faults. “Online” validation with averaging over 5 sequences/cycle.

	HCM2Neuro clusters		vector clusters	
	4 sensors	5 sensors	4 sensors	5 sensors
nominal weighting factor	1	1		
minimal fault grade			0.37	0.37
false-alarm rate	<b>0.0%</b>	<b>0.0%</b>	<b>0.0%</b>	<b>0.0%</b>

### 6.3.9 Simultaneous Faults

The fault diagnosis system was also trained with simultaneous faults. Therefore, data of the following nine fault cases were acquired, including some combinations of simultaneous faults (cf. Table 10 on page 116):

- single faults 1..5
- simultaneous faults 2 and 4
- simultaneous faults 2 and 5
- simultaneous faults 3 and 4
- simultaneous faults 3 and 5

A fault size of 30% was chosen for training. Note that some fault combinations are excluded. Fault 2 (reduced brine mass flow through the evaporator) and fault 3 (reduced evaporator heat transfer) both lower the pressure in the evaporator. A combination of these faults causes a low-pressure alarm and stops the machine automatically (due to the standard safety management). The combination of fault 4 (reduced water mass flow through the condenser) and fault 5 (reduced condenser heat transfer) causes a high-pressure alarm. Therefore, only the fault combination in the list above are trained.

First, a standard classification system is generated from the training data set, by using the HCM2Neuro clustering algorithm. One cluster is built for each

of the nine fault cases, including the fault combinations. Another data set was acquired for validation. Table 14 shows the classification results for the training data set as well as for the validation data set, in the case of four and five sensors (two and three output signals). With averaging, the rate of wrong classifications is reduced to zero for the fault size 30%. For the faults of size 20%, the rate of wrong classification is still low in the case of five sensors.

*Table 14: Rate of wrong classifications for HCM2Neuro clusters with single and simultaneous faults. The faults of size 30% are used for training, the faults of size 30% and 20% are used for validation. The number of sensors is either 4 or 5. Table cell contents: rate of wrong classifications.*

fault size	training data set, offline, no averaging		validation data set, offline, no averaging		validation data set, "online", averaging over 5 seq./cycle	
	4 sensors	5 sensors	4 sensors	5 sensors	4 sensors	5 sensors
20%					15.9%	2.3%
30%	<b>3.8%</b>	<b>3.4%</b>	<b>5.1%</b>	<b>4.0%</b>	<b>0%</b>	<b>0%</b>

The same data sets are used for training and checking vector clusters, where eight vectors represent each fault case (without the nominal case). The results are shown in Table 15. By using five sensors, the rate of wrong classifications is reduced to zero with averaging. In those cases, both the faults of size 30% and the faults of size 20% are classified well.

*Table 15: Rate of wrong classifications for vector clusters with single and simultaneous faults. The faults of size 30% are used for training, the faults of size 30% and 20% are used for validation. The number of sensors is either 4 or 5. Table cell contents: rate of wrong classifications.*

fault size	training data set, offline, no averaging		validation data set, offline, no averaging		validation data set, "online", averaging over 5 seq./cycle	
	4 sensors	5 sensors	4 sensors	5 sensors	4 sensors	5 sensors
20%					18.2%	0%
30%	<b>10.1%</b>	<b>3.4%</b>	<b>13.1%</b>	<b>5.9%</b>	<b>5.8%</b>	<b>0%</b>

## 7 Software FuzzyWatch

The diagnosis system of Chapter 5 is realized by a software tool written in MATLAB<sup>®</sup>. The software tool includes the training procedure as well as the online operation and provides a graphical user interface (GUI). The software version presented here has been adapted to the data of the IMRT test bench (cf. Section 6.3 and Chapter 12). The results of Section 6.3 were obtained by using this software version.

At startup, the software tool displays the main panel (Fig. 51), which contains the schematic diagram of the heat pump. First, the classification system has to be trained, as described in Section 7.1. This panel will then be used to display the classification results for online diagnosis (cf. Section 7.2).

### 7.1 Training Procedure

The training software needs three definition files containing information about the application under investigation:

- A file defining the model structure (according to the structure matrices of Section 5.2; for the IMRT test bench the full structure is defined)
- A file defining the names of the training data files which contain the measured data sequences ('cut data files'). A list contains all file names for each fault case ('fault file list'). Several lists can be defined, for example for various fault sizes.
- A file defining the measured signals, which are contained in the columns of the training data files.

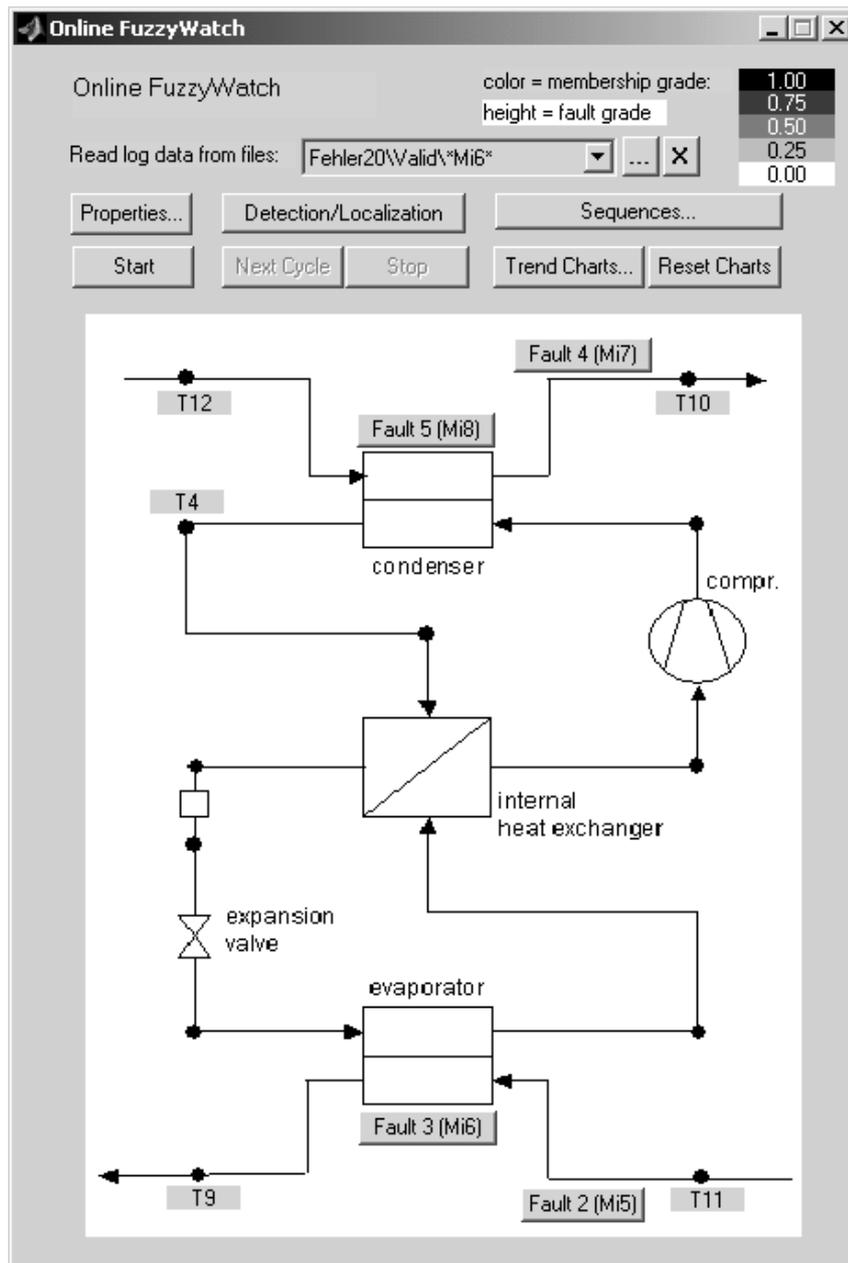


Fig. 51: Main panel.

By pressing the button 'properties' on the main panel and then the button 'train new' on the subsequent panel, the training panel of Fig. 52 is activated. On the schematic diagram of the heat pump, all sensors are displayed which are installed on the test bench. A subset of the signals can now be selected by the user from this initial sensor configuration. Here, the three output signals  $T_4$ ,  $T_9$ , and  $T_{10}$  are selected by the corresponding check boxes. The software tool automatically reduces the model structure defined above to the selected output signals by using the method of Section 5.5.2. The two input signals  $T_{11}$ ,  $T_{12}$  are required for all submodel selections.

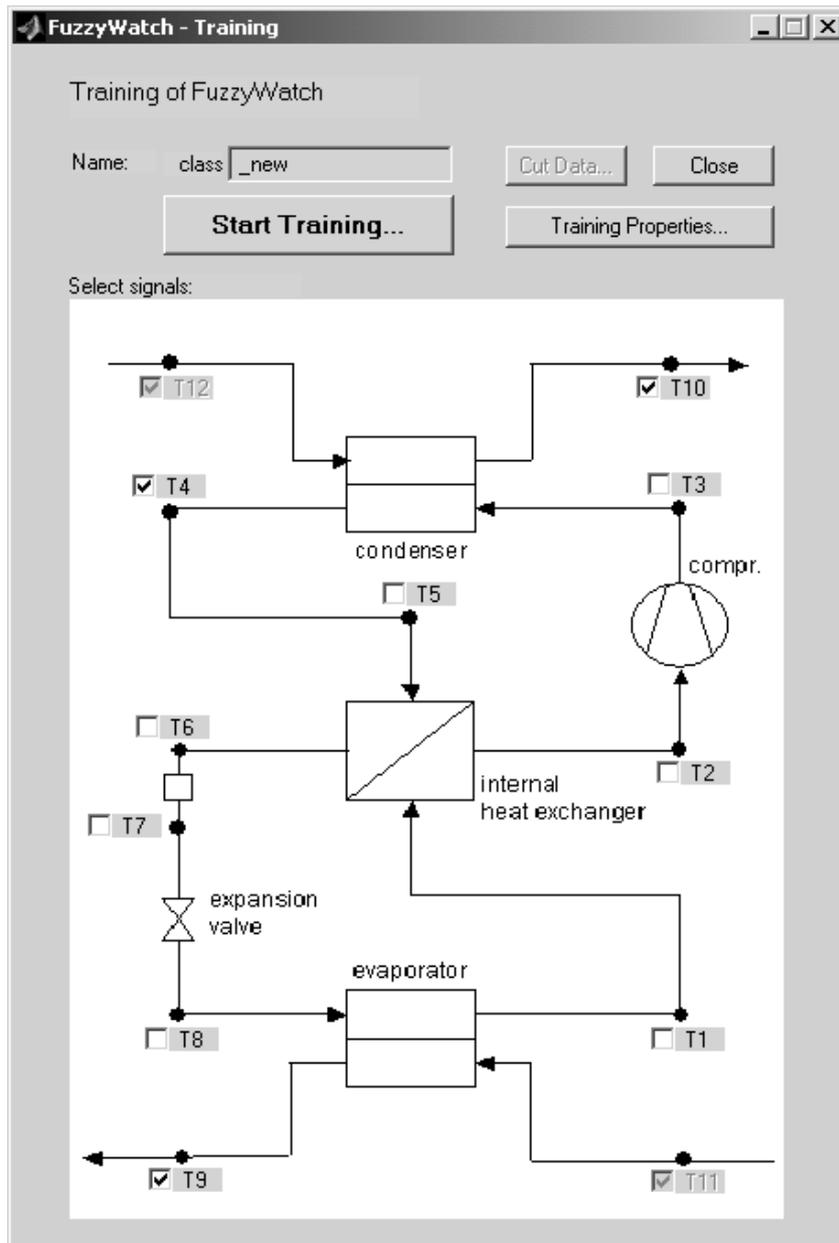
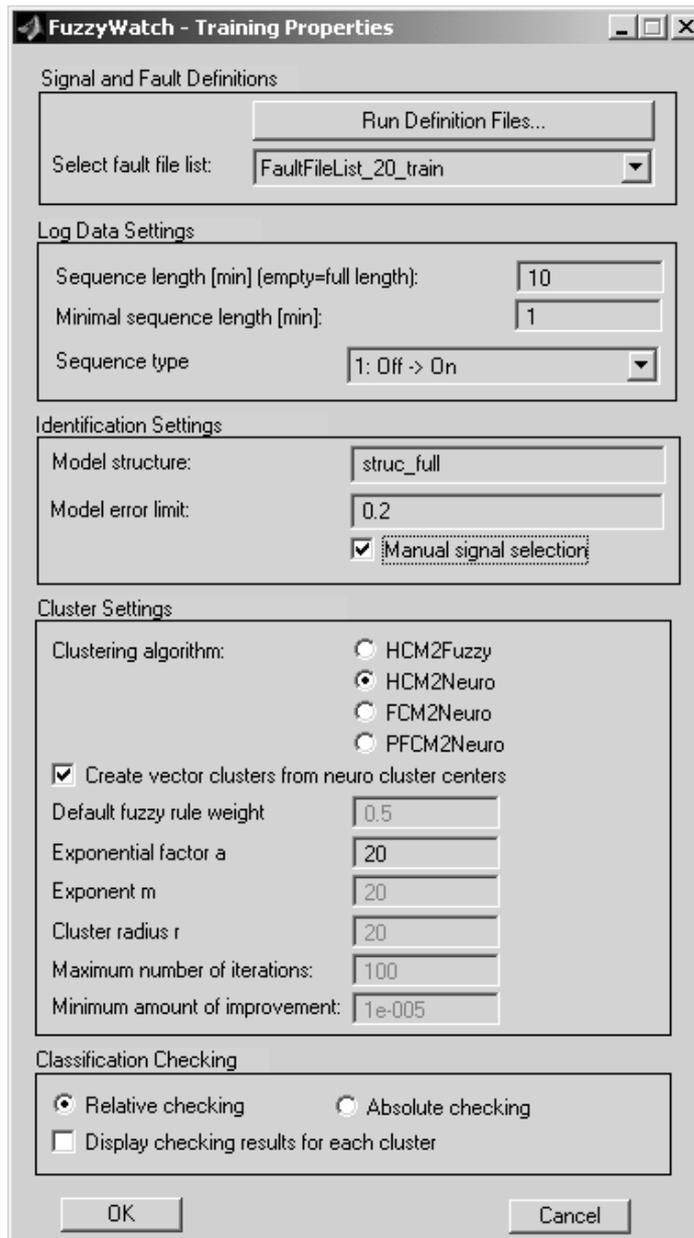


Fig. 52: User interface panel for training. The output signals are selected with check boxes and the classification system is automatically trained ('Start Training') and stored (in 'class\_new').

The properties for the training procedure are defined in Fig. 53. Pressing the button 'run definition files' causes the three definition files to be read. A list of training data files is selected ('select fault file list'), and the settings are defined for each training step. In the first paragraph, the length of the data sequences (in minutes), a minimal data sequence length and the sequence type ('off-on') is selected. Data sequences that are shorter than the minimal length are not evaluated at all. If the length of the data is chosen to be constant (here 10 minutes), then the data sequences are cut to this length.



*Fig. 53: Training properties.*

In the next paragraph, the model structure is selected. If the check box ‘manual signal selection’ is checked, the model error charts (cf. Section 6.3.4) are displayed and the user may select the appropriate submodels (or the corresponding output signals, respectively) and their model approaches. The software suggests the model approaches with the lowest errors. If the errors of both approaches are higher than a given limit (‘model error limit’, here 0.2), then the corresponding submodel is suggested to be eliminated. If the check box ‘manual signal selection’ is not checked, then the suggestions are automatically adopted.

In the next paragraph, one of the clustering algorithms of Section 5.4 is selected and the corresponding cluster parameters are defined (cf. eq. (26) on page 67, eq. (28) on page 68, eq. (32) on page 70). The additional cluster parameter ‘default fuzzy rule weight’ is used for the *HCM2Fuzzy* type. For classifying data points ‘outside’ of the trained clusters, an additional default rule is defined, which can be weighted. Usually all these parameters do not need to be changed, since the default values are chosen by experience. If the check box ‘create vector clusters from neuro cluster centers’ is selected, the vector clusters of Section 5.6.2 are created using the centers of the *HCM2Neuro*, *FCM2Neuro* or *PFCM2Neuro* clusters.

In the last paragraph, the settings for the classification checking algorithm are defined. By default, the checking algorithm described in Section 5.4.4 is used (‘relative checking’).

By pressing on the ‘Start Training’ button of the training panel (Fig. 52), the training steps are automatically executed:

- The model structure is reduced to the selected output signals.
- The submodel parameters are identified for each training data sequence of all fault cases. The model errors are then calculated for the different model approaches (static or 1<sup>st</sup> order ARX models).
- For each submodel, the best approach is selected by comparing the model errors. Submodels with high model errors are eliminated.
- Using the identified parameters, the fault clusters are built using the methods *HCM2Fuzzy*, *HCM2Neuro*, *FCM2Neuro*, or *PFCM2Neuro*.
- The vector clusters are optionally built from the standard cluster centers.
- The resulting classification system is checked by the training data.

After the training procedure, the resulting classification system is stored in a file (‘class\_new’ according to Fig. 52). It is then used for the online diagnosis described in the next section.

## 7.2 Online Diagnosis

After the training of the classification system, it is applied to the online data from operation. Therefore, the online fault diagnosis steps of Section 5.7.1 are executed repetitively for each data sequence.

### 7.2.1 Properties for Online Diagnosis

The classification system is selected from a list of trained systems (Fig. 54, ‘classification system’). For online diagnosis, the averaging method of Section 5.7.2 is used. Therefore, the number of averaged data sequences per cycle has to be defined in the ‘classification settings’. Additionally, the false-alarm tuning parameters, such as the nominal weighting factor for standard clusters and the minimal fault grade for vector clusters are defined (cf. Section 5.7.4). The maximum fault grade is used for scaling the fault grades (cf. Section 7.2.3) and trend charts (cf. Section 7.2.4).

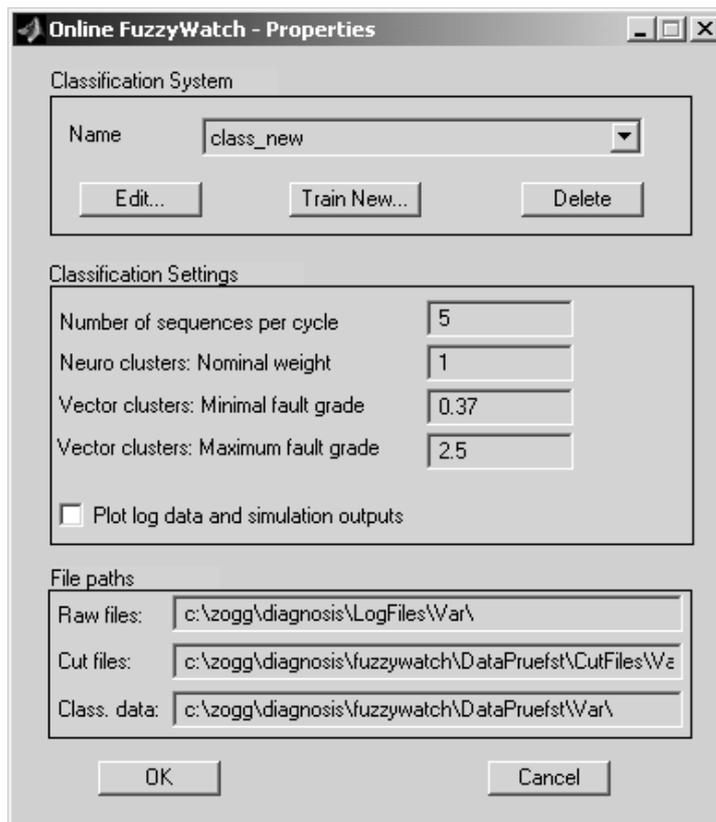


Fig. 54: Online diagnosis properties.

Now, the classification system is selected and the properties are defined. The next sections will show how the measured validation data will be classified sequentially for each cycle containing five data sequences each.

### 7.2.2 Sequential Classification with Standard Clusters

For the following evaluations, the same *HCM2Neuro* classification system and the same data sets are used as in Section 6.3.6. In Fig. 55, validation data of the fault case 3 (evaporator fouling, cf. Table 10 on page 116) is evaluated using the trained classification system.

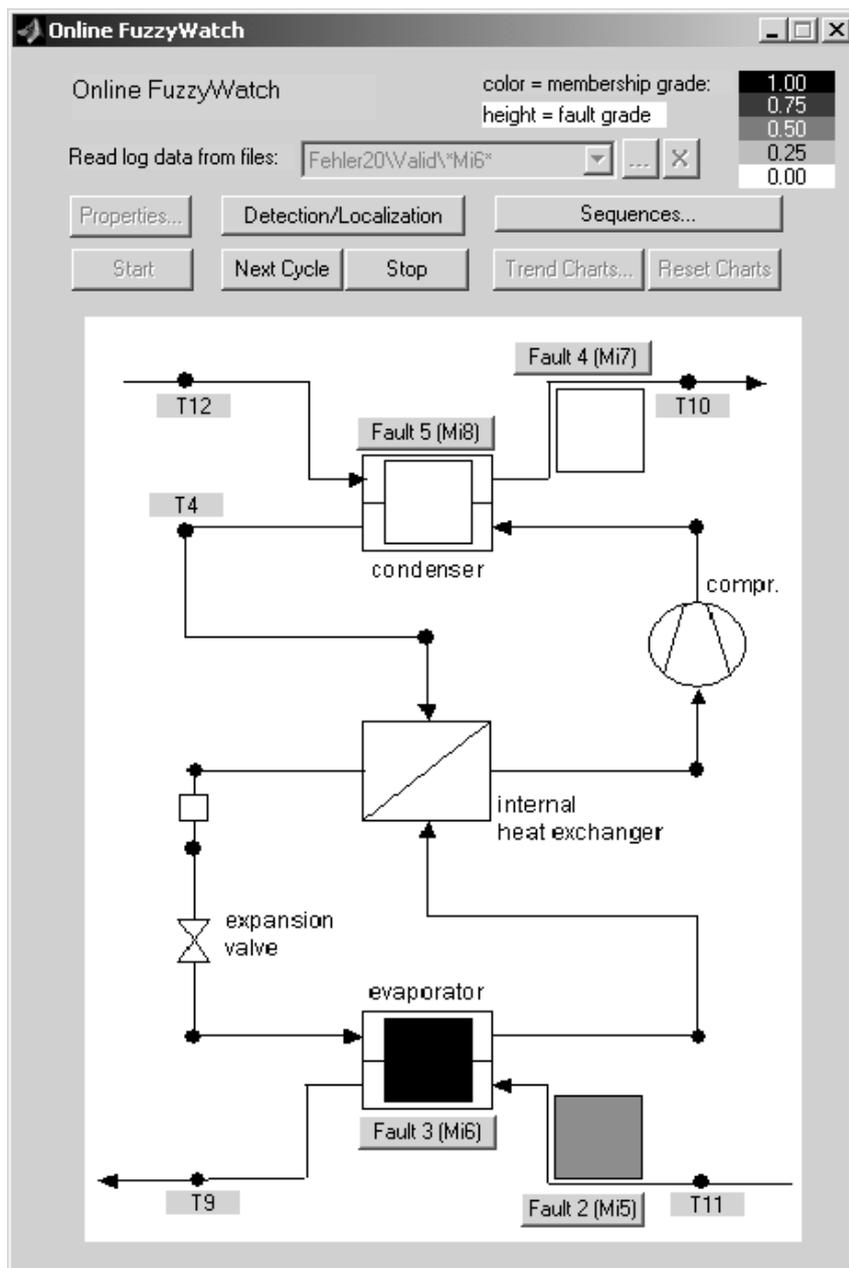


Fig. 55: HCM2Neuro classification results with validation data for fault 3, mean values of the membership grades for one cycle. White indicators for faults 4 and 5 = minimum membership grades (set to 0), black indicator for fault 3 = maximum membership grade (set to 1), gray indicator for fault 2 = low membership grade

Here, the membership grades are averaged over five sequences for one sample cycle. These grades are visualized with the shading of the indicators for each fault case (dark shadings for high membership grades). The indicators show that fault 3 is classified correctly with a maximum membership grade. Since faults 2 and 3 have similar effects on the signals, both fault cases have been 'activated', but fault 2 has a significantly lower membership grade.

Fig. 56 shows the membership grades for each data sequence of the cycle above. The five rows correspond to the fault cases while the five columns correspond to the sequences. The additional column displays the mean values. Again, dark shadings indicate high membership grades. With one exception, all sequences have maximum membership grades for fault 3. Only one sequence is classified wrong (“outlier”), but with regard to the mean values, the classification is correct.

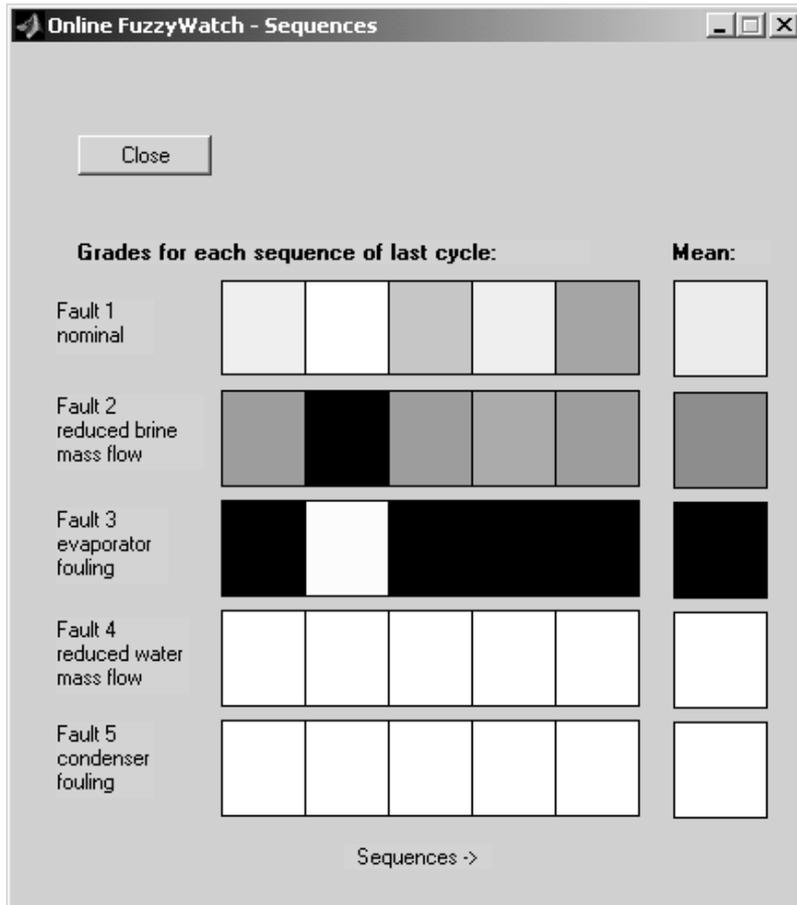


Fig. 56: HCM2Neuro classification results with validation data for fault 3, membership grades for each sequence (5 sequences, 5 columns) and mean values of the whole cycle (last column). White indicators = minimum membership grades (set to 0), black indicators = maximum membership grades (set to 1), gray indicators = membership grades inbetween

### 7.2.3 Sequential Classification with Vector Clusters

For the following evaluations, the same vector classification system and the same data sets are used as in Section 6.3.7. In Fig. 57, the validation data of fault case 3 (evaporator fouling) is evaluated using the trained classification system.

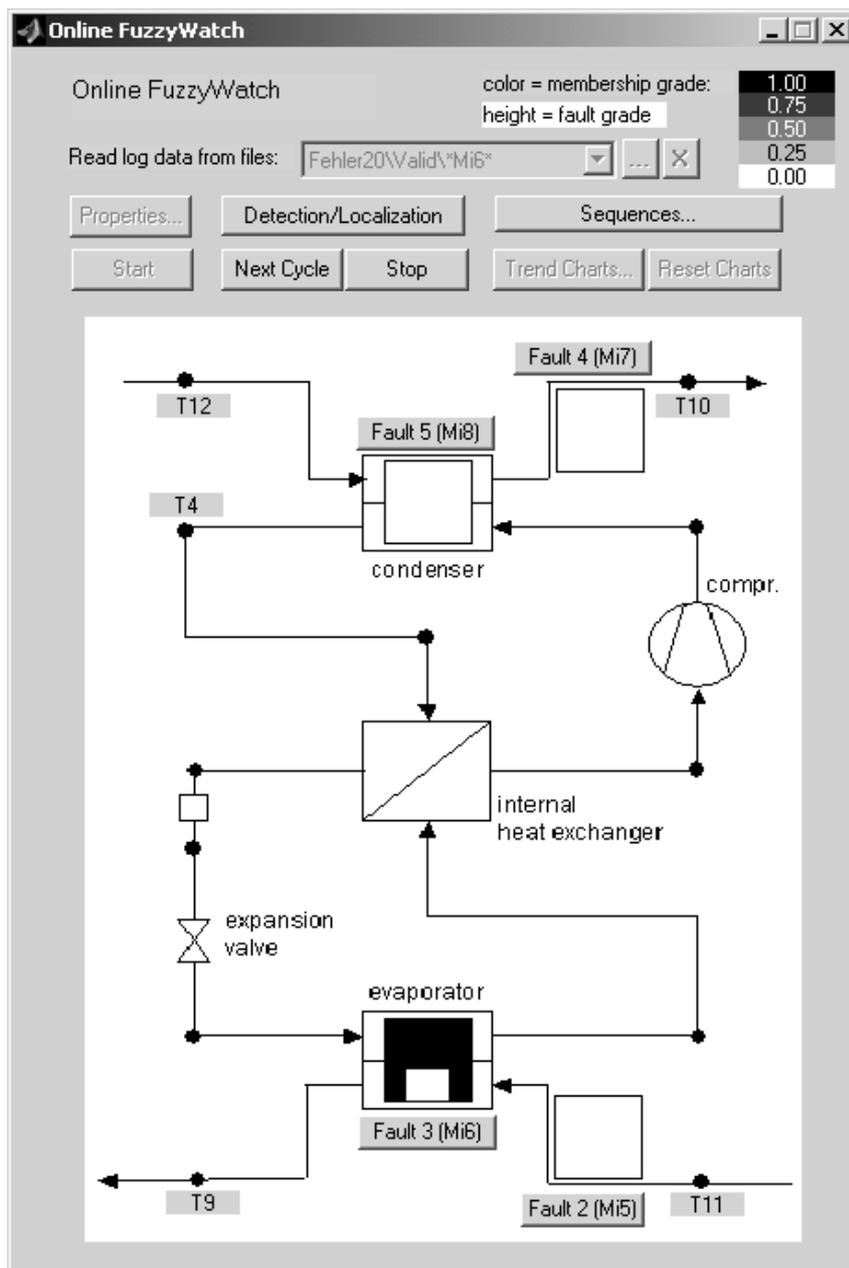


Fig. 57: Vector classification results with validation data for fault 3, mean values of the membership grades and fault grades for one cycle. White indicators for faults 2, 4, and 5 = minimum membership grades (set to 0), black indicator for fault 3 = maximum membership grade (set to 1), height of the white bar for fault 3 = fault grade 0.96

The membership grades as well as the fault grades are averaged over five sequences for the same sample cycle. The membership grades are visualized with the shading of the indicators (dark indicators for high membership grades), whereas the fault grades are visualized with white bars in the foreground (cf. fault 3 in Fig. 57). The height of the white bar displays the size of the fault.

The fault diagnosis system directly locates the faults. Nevertheless, a two-step display with a fault detection view and a fault localization view is useful in practice. By pressing the button “Detection/Localization”, the view can be changed (Fig. 58).

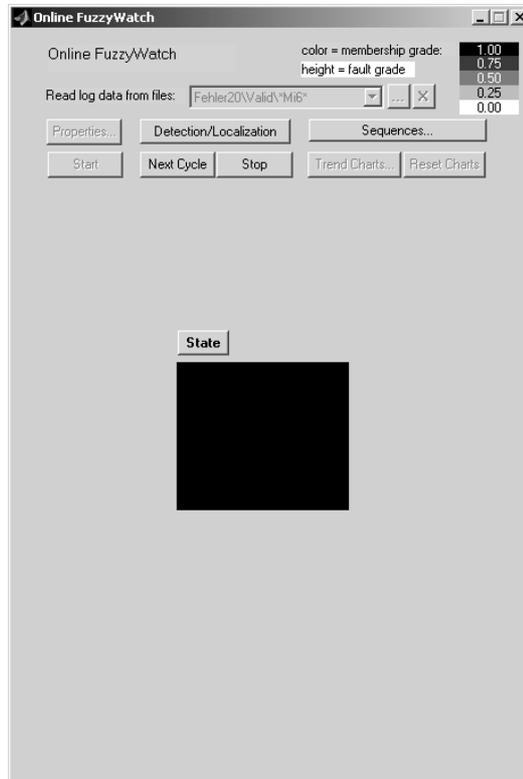


Fig. 58: Classification results with validation data for fault 3, fault detection view. ‘State’ is black = there is a fault present.

In the detection view, the ‘state’ of the machine indicates whether or not there is any fault present (dark indicator). The ‘state’ is given by the maximum of the membership grades for the current cycle (according to the mean values of Fig. 57, here it is given by the membership grade of fault 3).

Fig. 59 shows the membership grades (shadings) as well as the fault grades (height of the white bars in the foreground) for each data sequence of the cycle above. A fault grade of 1 means that the current fault has the same size as the fault that has been trained. The rows 2..5 correspond to the fault vectors while the five columns correspond to the sequences. The first row with the nominal case is not used here, because the fault vectors are only defined for the fault cases 2..5 (cf. Section 5.6.2).

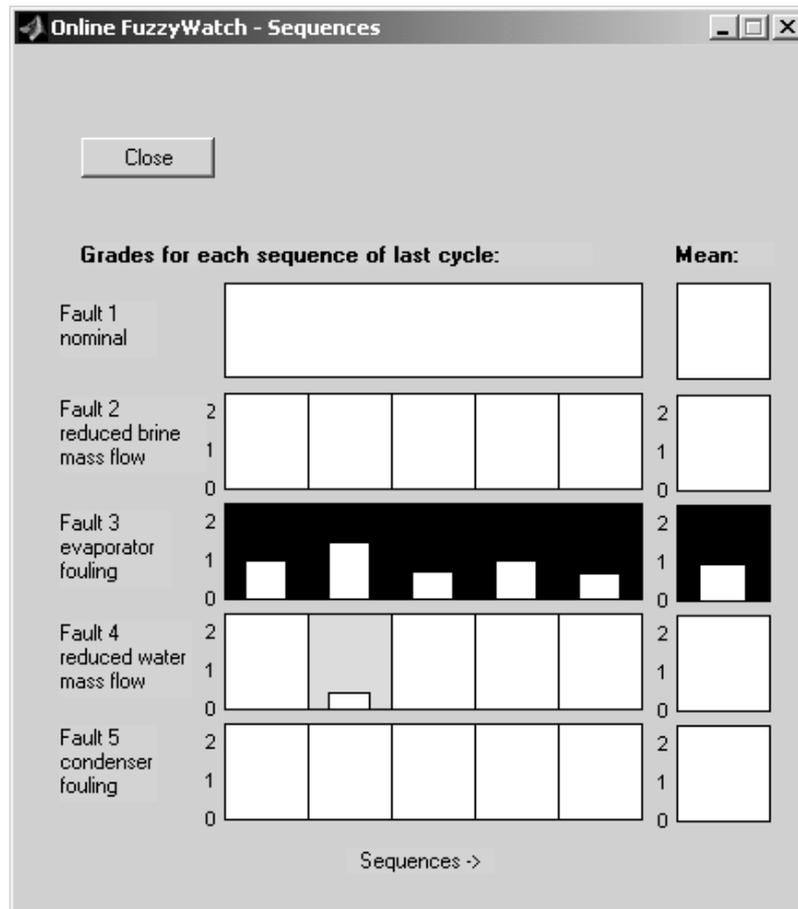


Fig. 59: Vector classification results with validation data for fault 3, membership grades and fault grades for each sequence (5 sequences in 5 columns) and mean values of the whole cycle (last column). White indicators = minimum membership grades (set to 0), black indicators = maximum membership grades (set to 1), height of the white bars = fault grades (axis labels on the left)

In the additional column, the mean values are displayed. Here the mean values as well as all sequences have maximum membership grades for fault 3 (dark shadings). Thus, the mean values as well as all sequences are correctly classified. With one exception, all other membership grades are set to zero, because their fault grades are lower than the minimal fault grade (cf. Section 5.7.4).

### 7.2.4 Trend Charts

The progression of the faults are visualized with trend charts (Fig. 60), where the mean values of each consecutive cycle are plotted. Here, an increasing size of fault 5 (condenser fouling) is recognizable, while fault 4 (reduced water mass flow) is only slightly ‘activated’. For plotting the charts, the fault grades (fault sizes) and the membership grades (fault probabilities 0..1) are combined in one graph by multiplying them. Optionally, they can be displayed in two graphs. The maximum fault grade defines the range of the fault size axis, whereas a fault grade of 1 corresponds to the size of the trained fault (maximum fault grade = 2.5, cf. settings shown in Fig. 54).

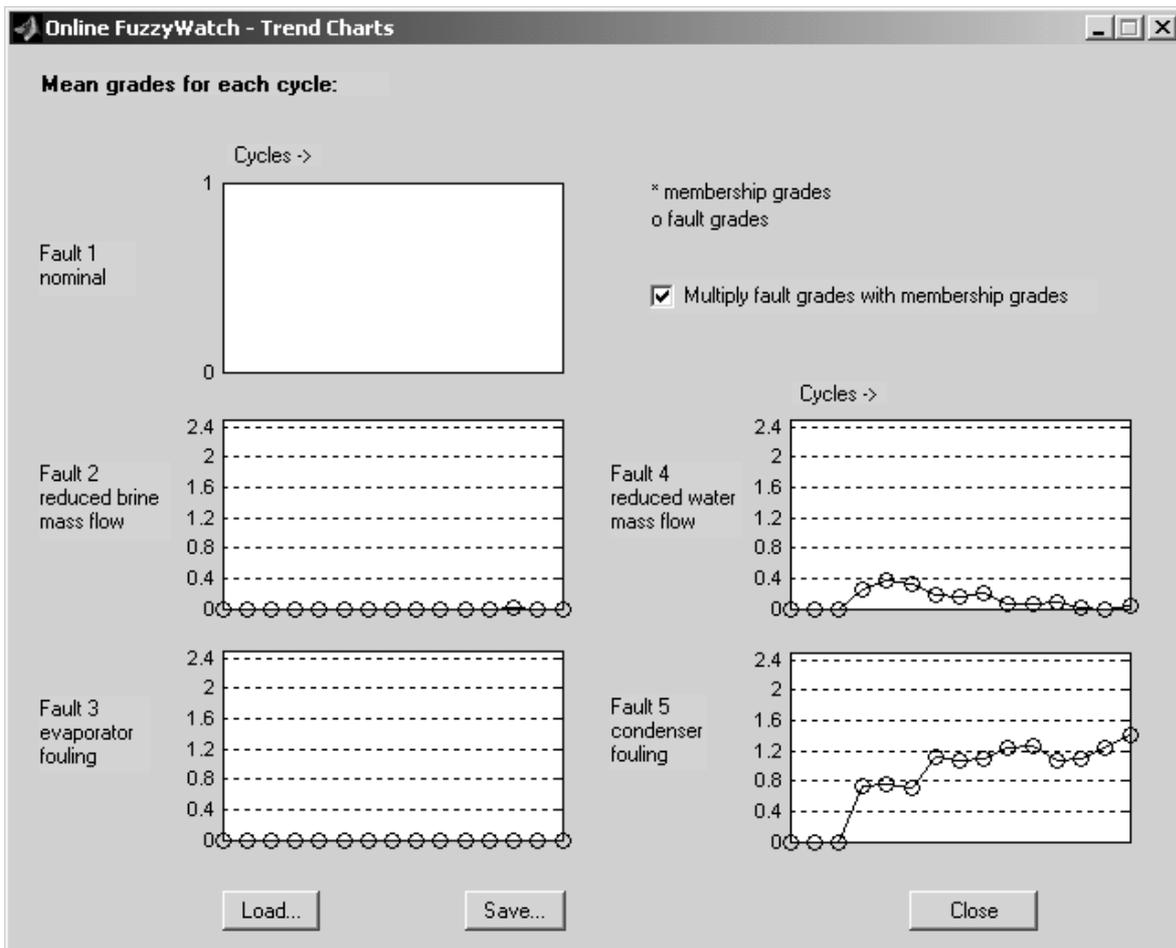


Fig. 60: Trend charts for validation data of fault 5 with an increasing fault size (axis labels on the left). Each point (circle) corresponds to one cycle mean value, the fault grades are multiplied by the membership grades.

## 8 A Fault Diagnosis System with Steady-State Parameter Identification – Theory

This diagnosis system is based on the identification of the parameters and characteristics of the machine (Fig. 61). Temperature signals, pressure signals and an additional electric power signal are measured. No mass flow sensors or heat flow sensors are needed, which helps to reduce the costs.

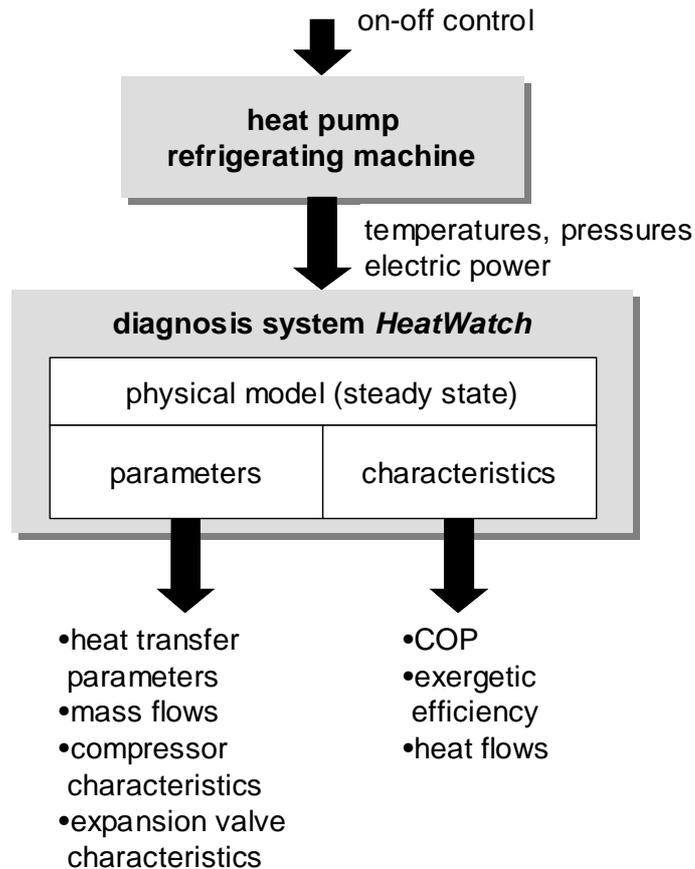


Fig. 61: The schematic representation of the fault diagnosis system called HeatWatch.

### 8.1 Concept

During the operational phase, the diagnosis system of this chapter identifies the current parameters and compares them to certain reference parameters (Fig. 62, right side and Section 8.1.2). The reference parameters are identified in an initialization step before or at the beginning of the operational phase (Fig. 62, left side and Section 8.1.1). Section 14.1.2 describes the details as to when this initialization step has to be carried out. Otherwise no training is needed, since no classification of the faults is trained. In particular, no data of any fault case is required for training.

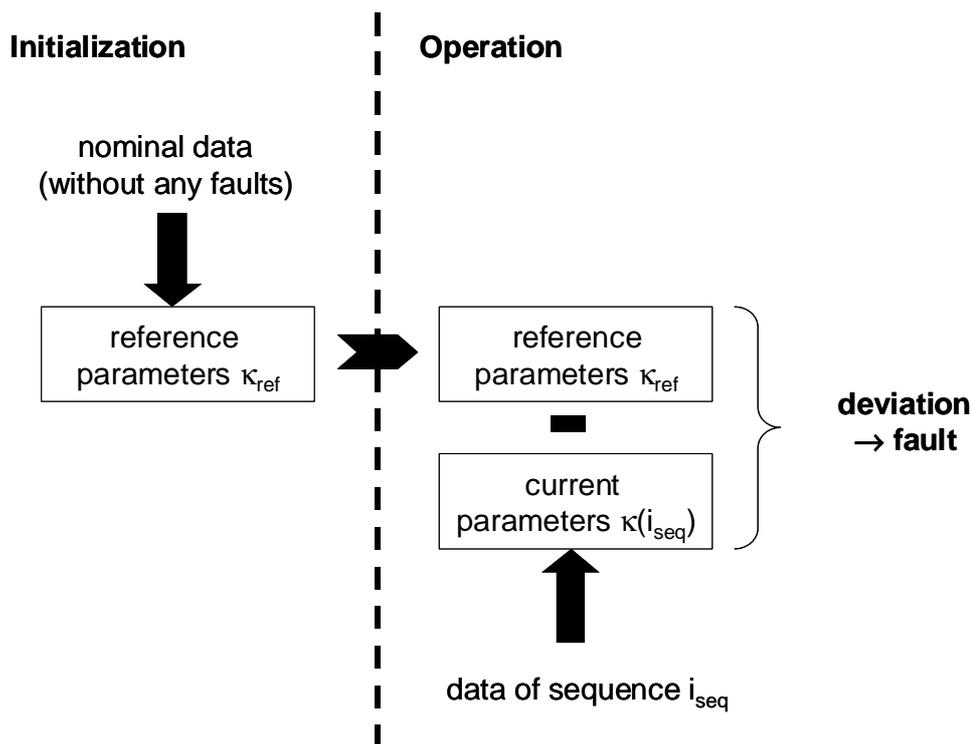


Fig. 62: Initialization and Operation

### 8.1.1 Initialization

The same procedure as in Section 8.1.2 is used to identify a reference parameter set  $\kappa_{ref}$  from nominal (fault-free) data. Therefore, the heat pump has to be run at one operating point until a (quasi) steady state is reached. In order to improve the reliability, more than one parameter set may be identified at the same operating point or at varying operating points. These parameter sets are then averaged for defining the reference parameter set.

### 8.1.2 Operation

During operation, the heat pump is switched on and off, which results in the data sequences  $i_{seq}$  with a duration in the range of 10 minutes up to several hours. A sample signal  $y_i$  for one on sequence is shown in Fig. 63. For diagnosis, the steady-state value of the signal is used only. Since the steady state is not reached within a data sequence of finite length, the quasi steady-state value  $y_{i,ss}$  is evaluated instead. For the evaluation in steady state, the current operating mode is important only (here ‘on’), but not the operating mode in the past (here ‘off’). Therefore the sequence is only specified by the current operating mode (here ‘on sequence’).

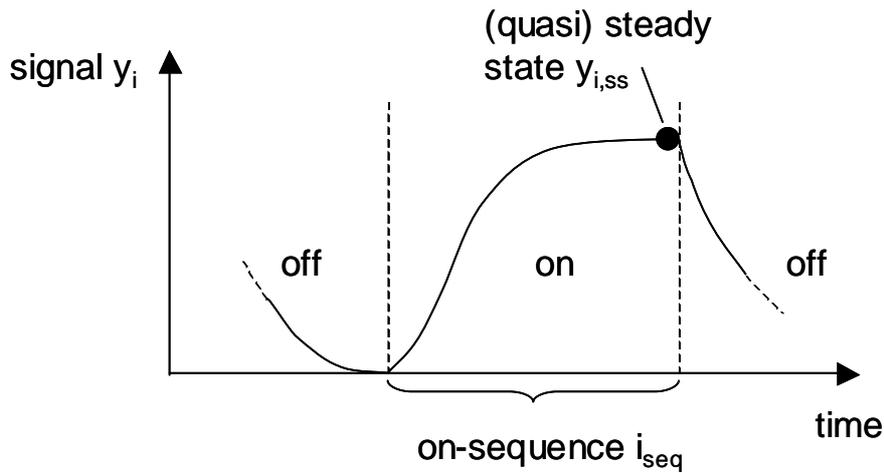


Fig. 63: Step response and steady-state value for one signal during the sequence  $i_{seq}$

The (quasi) steady-state values  $y_{i,ss}(i_{seq})$  of all measured signals are detected and used to identify the parameter set  $\kappa(i_{seq})$  of a steady-state heat pump model (cf. Section 8.2). The parameter set  $\kappa(i_{seq})$  is compared to the reference parameter set  $\kappa_{ref}$ . From the parameter deviations, the faults can be interpreted by the user (cf. Section 8.5).

## 8.2 Modeling and Parameter Identification

The model has been developed for a heat pump with an internal heat exchanger (Fig. 64), which can either be an air-to-water heat pump or a brine-to-water heat pump. The equations of this section are valid for all types of heat sources. Here the normal operating mode ('on') is handled only. Therefore no hot gas defrosting cycle has to be modelled in the case of an air-to-water heat pump.

### 8.2.1 Sensors Required

Due to the physical steady-state model presented in the next section, seven temperature signals ( $T_i$ ), two pressure signals ( $p_i$ ), and one power signal ( $P_{HP}$ ) has to be measured (list (56)). The locations of the corresponding sensors are displayed in Fig. 64.

$$\left[ T_{ov} \quad T_{hg} \quad T_{sc} \quad T_{s,i} \quad T_{s,o} \quad T_{w,i} \quad T_{w,o} \quad p_{lp} \quad p_{hp} \quad P_{HP} \right] \quad (56)$$

In standard commercial heat pumps, four temperature sensors  $T_{s,i}$ ,  $T_{s,o}$  and  $T_{w,i}$ ,  $T_{w,o}$  at the heat source side and at the water side are already installed. The other temperature sensors  $T_{ov}$  (superheating temperature),  $T_{hg}$  (hot gas temperature), and  $T_{sc}$  (subcooling temperature) have to be mounted in the refrigerant

erant cycle. Additionally, two pressure sensors are required, which measure the low pressure signal  $p_{lp}$  and the high pressure signal  $p_{hp}$ . Here, these pressures are assumed to be measured before and after the compressor. Finally, the electric power consumption of the compressor is measured in order to calculate the refrigerant mass flow.

These eleven signals will be used in the steady-state equations of the next section. Only their values in (quasi) steady state are thus used. A method for detecting steady state is presented in Section 8.3

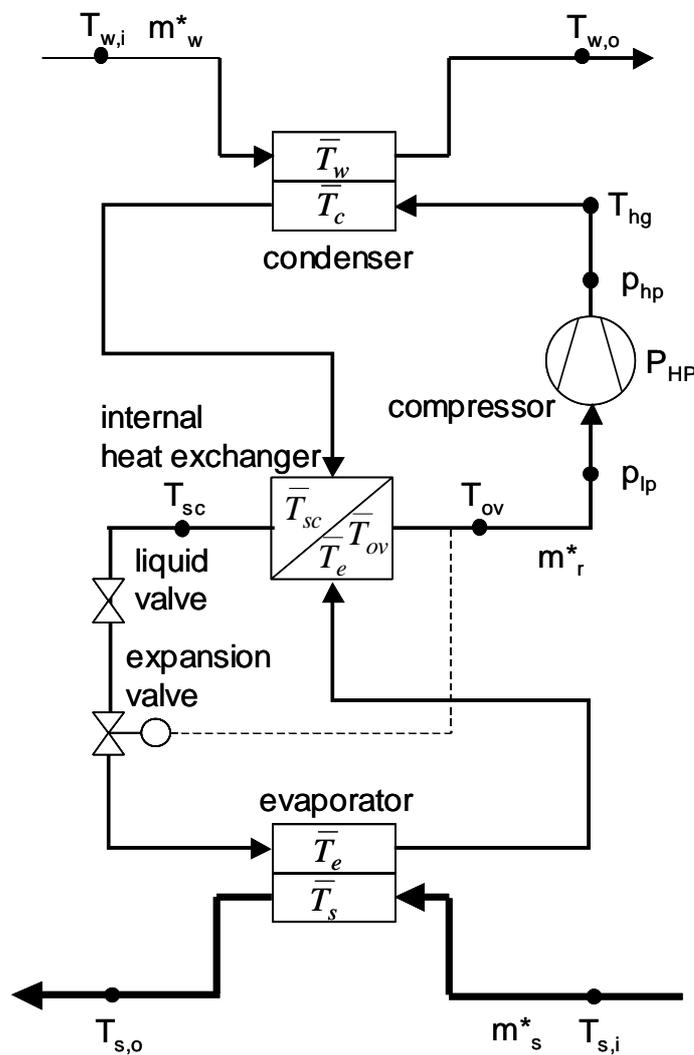


Fig. 64: A heat pump with an internal heat exchanger.

### 8.2.2 Steady-State Equations

The model is directly built for steady-state condition. The energy balances are defined for the evaporation process in the evaporator and the internal heat exchanger (eq. (57)), for the superheating process in the internal heat exchanger (eq. (58)), for condensation (eq. (59)), for the subcooling process in the internal heat exchanger (eq. (60)), for the heat source side (eq. (61)), and for the heat sink side (eq. (62)). All terms represent heat flows.

$$m_r^* \cdot r(p_{lp}) = kA_e(\overline{T}_s - \overline{T}_e) + kA_{he,e}(\overline{T}_{sc} - \overline{T}_e) \quad (57)$$

$$m_r^* \cdot \overline{c}_{ov} \cdot (T_{ov} - T_d(p_{lp})) = kA_{he,ov}(\overline{T}_{sc} - \overline{T}_{ov}) \quad (58)$$

$$m_r^* \cdot r(p_{hp}) = kA_c(\overline{T}_c - \overline{T}_w) \quad (59)$$

$$m_r^* \cdot \overline{c}_{sc} \cdot (T_b(p_{hp}) - T_{sc}) = kA_{he,ov}(\overline{T}_{sc} - \overline{T}_{ov}) + kA_{he,e}(\overline{T}_{sc} - \overline{T}_e) \quad (60)$$

$$kA_e(\overline{T}_s - \overline{T}_e) = m_s^* \cdot \overline{c}_s \cdot (T_{s,i} - T_{s,o}) \quad (61)$$

$$kA_c(\overline{T}_c - \overline{T}_w) = m_w^* \cdot \overline{c}_w \cdot (T_{w,o} - T_{w,i}) \quad (62)$$

All the temperature signals  $\overline{T}_i$  are measured by the sensors of Section 8.2.1, and the mean temperatures  $T_i$  are calculated from the measured signals (see below). Also the low pressure signal  $p_{lp}$  and the high pressure signal  $p_{hp}$  are measured. The mass flows, such as the refrigerant mass flow  $m_r^*$ , the source mass flow  $m_s^*$ , and the water mass flow  $m_w^*$ , are not measured. The heat transfer parameters  $kA_i$ , which are the product of the heat transfer coefficient  $k_i$  and the heat transfer area  $A_i$ , are unknown.

The latent heat of vaporization  $r(p_i)$ , the bubble-point temperature  $T_b(p_i)$ , and the dew-point temperature  $T_d(p_i)$  are given by the refrigerant data equations (cf. Section 11.1.11). The specific heat capacities  $\overline{c}_i$  are either calculated from refrigerant data or regarded as parameters. For calculating the mean temperatures, the arithmetic mean values are used (eqs. (63) to (66)).

$$\overline{T}_{ov} = \frac{T_d(p_{lp}) + T_{ov}}{2} \quad (63)$$

$$\overline{T}_{sc} = \frac{T_b(p_{hp}) + T_{sc}}{2} \quad (64)$$

$$\overline{T_s} = \frac{T_{s,i} + T_{s,o}}{2} \quad (65)$$

$$\overline{T_w} = \frac{T_{w,i} + T_{w,o}}{2} \quad (66)$$

The evaporation and condensation mean temperatures are calculated from refrigerant data, using the mid-point temperatures  $T_m(p_i)$ .

$$\overline{T_e} = T_m(p_{lp}) \quad (67)$$

$$\overline{T_c} = T_m(p_{hp}) \quad (68)$$

Since the six steady-state equations (57) to (62) have seven unknown parameters  $m_i^*$  and  $kA_i$ , one parameter has to be calculated first. Here the refrigerant mass flow  $m_r^*$  is calculated from the power input  $P_{HP}$  of the compressor (eq. (71)) by using the difference between the hot gas enthalpy  $h_{hg}$  (eq. (69) with the hot gas temperature  $T_{hg}$  and the high pressure  $p_{hp}$ ) and the superheating enthalpy  $h_{ov}$  (eq. (70) with the superheating temperature  $T_{ov}$  and the low pressure  $p_{lp}$ ). The electro-mechanical efficiency  $\eta_{cmp}$  may be regarded as a scaling factor for the mass flow  $m_r^*$ , thus it is sufficient to use an approximative value.

$$h_{hg} = h''(p_{hp}) + \overline{c_{hg}} \cdot (T_{hg} - T_d(p_{hp})) \quad (69)$$

$$h_{ov} = h''(p_{lp}) + \overline{c_{ov}} \cdot (T_{ov} - T_d(p_{lp})) \quad (70)$$

$$m_r^* = \frac{\eta_{cmp} P_{HP}}{h_{hg} - h_{ov}} \quad (71)$$

Alternatively, the mass flow  $m_r^*$  could be calculated from the pressure rise ( $p_{hp}-p_{lp}$ ) over the compressor. But then the mass flow characteristics of the compressor would be needed, which are not known exactly because of the rather high variations in the data of individual compressors of the same type. By measuring the power input, the compressor characteristics are not used.

### 8.2.3 Linear Regression Form

With the known refrigerant mass flow  $m_r^*$ , the steady-state equation system (57)..(62) is transformed into the linear regression form (eq. (72)), with the known output vector  $y$  (eq. (73)), the known regression matrix  $\varphi$  (eq. (74)), and the unknown parameter vector  $\kappa$  (eq. (73)).

$$y = \varphi \cdot \kappa \quad (72)$$

$$y = \begin{bmatrix} m_r^* \cdot r(p_{lp}) \\ m_r^* \cdot \overline{c_{ov}} \cdot (T_{ov} - T_d(p_{lp})) \\ m_r^* \cdot r(p_{hp}) \\ m_r^* \cdot \overline{c_{sc}} \cdot (T_b(p_{hp}) - T_{sc}) \\ 0 \\ 0 \end{bmatrix} \quad \kappa = \begin{bmatrix} kA_e \\ kA_{he,ov} \\ kA_c \\ kA_{he,e} \\ m_s^* \\ m_w^* \end{bmatrix} \quad (73)$$

$$\varphi = \begin{bmatrix} \overline{T_s} - \overline{T_e} & 0 & 0 & \overline{T_{sc}} - \overline{T_e} & 0 & 0 \\ 0 & \overline{T_{sc}} - \overline{T_{ov}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \overline{T_c} - \overline{T_w} & 0 & 0 & 0 \\ 0 & \overline{T_{sc}} - \overline{T_{ov}} & 0 & \overline{T_{sc}} - \overline{T_e} & 0 & 0 \\ \overline{T_e} - \overline{T_s} & 0 & 0 & 0 & \overline{c_s}(T_{s,i} - T_{s,o}) & 0 \\ 0 & 0 & \overline{T_w} - \overline{T_c} & 0 & 0 & \overline{c_w}(T_{w,o} - T_{w,i}) \end{bmatrix} \quad (74)$$

The parameters  $\kappa$  are identified therefrom in the next section.

### 8.2.4 Parameter Identification

The parameters are identified using the (quasi) steady-state values of the signals. Before evaluation, the signals are averaged over a certain number of samples during a so-called ‘steady-state interval’ in order to minimize the effect of noise. Thus, for calculating all variables in  $\varphi$  (eq. (74)) and  $y$  (eq. (73)), the mean values of the signals are used, which results in eq. (75). The matrix  $\overline{\varphi}$  is then inverted in order to identify the parameters  $\kappa$  (eq. (76)).

$$\overline{y} = \overline{\varphi} \cdot \kappa \quad (75)$$

$$\hat{\kappa} = \overline{\varphi}^{-1} \overline{y} \quad (76)$$

The detection of the steady-state interval will be discussed in the next section.

## 8.3 Steady-State Detection

The steady state is either detected by calculating the derivative of a characteristic signal or by using a fixed steady-state time.

**Detection by calculating the derivative.** A characteristic signal  $x$  with a slow transient is chosen for steady-state detection. For the application I in Section 9.1, the hot gas temperature  $T_{hg}$  is chosen (cf. Fig. 65). Apparently, the signal is superposed with oscillations, due to the ‘hunting effect’ of the thermostatic expansion valve.

The steady state is determined by the derivative  $dx/dt$  of the detection signal  $x$ . A signal containing disturbances (as the oscillations in  $T_{hg}$ ) or noise first has to be filtered (eq. (77)), whereas for that application a low-pass Butterworth filter of order 3 is selected (eq. (78)) (for time-discrete systems, cf. [Geering 01]). The system is in (quasi) steady state, if the derivative  $dx_f/dt$  of the filtered signal  $x_f$  is lower than a given limit  $dx_{limit}$  during the time interval  $(t_{ss} \dots t_{ss} + \Delta t_{ss})$ .

$$X_f(z) = H(z) \cdot X(z) \quad (77)$$

$$H(z) = \frac{b_1 + b_2 z^{-1} + b_3 z^{-2} + b_4 z^{-3}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}} \quad (78)$$

$$\frac{dx_f(t)}{dt} \leq dx_{limit} \quad t \in (t_{ss} \dots t_{ss} + \Delta t_{ss}) \quad (79)$$

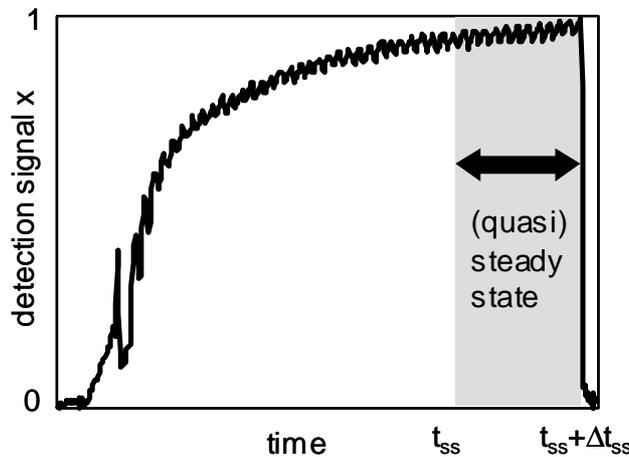


Fig. 65: Detection signal and (quasi) steady-state interval. Sample signal: hot gas temperature.

**Fixed steady-state time.** A much simpler approach works with a fixed steady-state time  $t_{ss}$ . With this method, the filtering and the calculation of the derivative is omitted and the results are even more reliable. But the steady-state start time  $t_{ss}$  has to be chosen very carefully by observing relevant data sequences before.

Typically, the duration of the steady-state interval  $\Delta t_{ss}$  is about 10%-30% of the length of the data sequences, depending on the duration of the transient part of the signals. Especially if there are oscillations in the signals and if the signals are not filtered, the interval should include several periods of these oscillations.

## 8.4 Additional Parameters and Characteristics

From the identified parameters  $\hat{\kappa}$  (eq. (76)) and the measured signals, additional parameters and characteristics of the heat pump are calculated. The parameters of the compressor, such as the polytropic exponent  $c_{cmp}$  and the mass flow supply efficiency  $\lambda_{cmp}$ , are calculated by eqs. (80) and (81), with  $n_{cmp}$  as the given compressor speed and  $b_{cmp}$  as a given characteristic parameter.

$$c_{cmp} = \frac{\log(T_{hg} / T_{ov})}{\log(p_{hp} / p_{lp})} \quad (80)$$

$$\lambda_{cmp} = \frac{m_r^*}{n_{cmp} \cdot (p_{hp} / p_{lp})^{b_{cmp}}} \quad (81)$$

The valve constant  $k_{exp}$  (eq. (82)) and the superheating  $\Delta T_{ov}$  (eq. (83)) characterize the expansion valve.

$$k_{exp} = \frac{m_r^*}{\sqrt{p_{hp} - p_{lp}}} \quad (82)$$

$$\Delta T_{ov} = T_{ov} - T_d(p_{lp}) \quad (83)$$

All the heat flows that are represented by the terms in the system (57)..(62) can be calculated, if the parameters  $kA_i$  and the mass flows  $m_i^*$  are known. Eqs. (84) and (85) show two examples for the heat-source heat flow  $Q_s^*$  and for the heat-output heat flow  $Q_{HP}^*$ , which are calculated by using one of the corresponding terms.

$$Q_s^* = m_s^* \bar{c}_s (T_{s,i} - T_{s,o}) = kA_e (\bar{T}_s - \bar{T}_e) \quad (84)$$

$$Q_{HP}^* = Q_w^* = m_w^* \bar{c}_w (T_{w,o} - T_{w,i}) = kA_c (\bar{T}_c - \bar{T}_w) = m_r^* r(p_{hp}) \quad (85)$$

For the overall energy balance of the heat pump, condition (86) is to be fulfilled. By expressing  $Q_{HP}^*$ ,  $P_{HP}$ , and  $Q_s^*$  in terms of  $m_r^*$  and then dividing by

$m_r^*$ , this condition leads to eq. (87). Thus, the heat capacities  $\overline{c_{sc}}$  and  $\overline{c_{ov}}$  cannot be chosen arbitrarily, but they have to fulfill eq. (87). Here,  $\overline{c_{ov}}$  is calculated from refrigerant data and eq. (87) is solved for  $\overline{c_{sc}}$ .

$$Q_{HP}^* = \eta_{cmp} P_{HP} + Q_s^* \quad (86)$$

$$r(p_{hp}) = (h_{hg} - h_{ov}) + r(p_{lp}) - \overline{c_{sc}} \cdot (T_b(p_{hp}) - T_{sc}) + \overline{c_{ov}} \cdot (T_{ov} - T_d(p_{lp})) \quad (87)$$

Other heat pump characteristics, such as the coefficient of performance  $COP_{HP}$  (eq. (88)) and the exergetic efficiencies  $\varepsilon_i$  (eq. (90), [Kuijk 96]), are also calculated. The Carnot coefficient  $COP_{Carnot,i}$  is either determined by the source and water temperatures ( $T_s$ ,  $T_w$ , index *sw*), or it is determined by the evaporator and condenser temperatures ( $T_e$ ,  $T_c$ , index *ec*).

$$COP_{HP} = \frac{Q_{HP}^*}{P_{HP}} \quad (88)$$

$$COP_{Carnot,sw} = \frac{T_w}{T_w - T_s} \quad COP_{Carnot,ec} = \frac{T_c}{T_c - T_e} \quad (89)$$

$$\varepsilon_{sw} = \frac{COP_{HP}}{COP_{Carnot,sw}} \quad \varepsilon_{ec} = \frac{COP_{HP}}{COP_{Carnot,ec}} \quad (90)$$

Using the mass flow  $m_w^*$  from steady state, the heat output  $Q_{HP}^*(t)$  can also be estimated for the time interval before steady state (eq. (91)). During the data sequences the mass flow  $m_w^*$  through the circulation pump is almost constant.

$$Q_{HP}^*(t) = m_w^* \overline{c_w} (T_{w,o}(t) - T_{w,i}(t)) \quad (91)$$

Thus, the energetic efficiency  $PF$  is given by eq. (92) for a sequence of the duration  $\Delta t_{seq}$ .

$$PF(\Delta t_{seq}) = \frac{\int_{t=0}^{\Delta t_{seq}} Q_{HP}^*(t) dt}{\int_{t=0}^{\Delta t_{seq}} P_{HP}(t) dt} \quad (92)$$

## 8.5 Interpretation of the Parameters

From the deviations between the current physical parameters  $\kappa$  and the reference parameters  $\kappa_{ref}$ , the corresponding faults are interpreted by the user. Table 16 summarizes the interpretations of the changed parameters. There are eleven parameters that are used for fault diagnosis. The characteristics  $\varepsilon_{sw}$  and  $\varepsilon_{ec}$  in the last row are used for fault detection.

Table 16: Changed parameters of the steady-state model and corresponding faults.

changed parameter(s)	parameter description	fault interpretation (in brackets: for air-to-water heat pump)
$kA_e$	evaporator heat transfer	evaporator fouling (icing)
$kA_{he,e}$ , $kA_{he,ov}$	internal heat exchanger: evaporation heat transfer, superheating heat transfer	internal heat exchanger fouling
$kA_c$	condenser heat transfer	condenser fouling
$m_w^*$	water mass flow	water cycle fouling, water pump malfunction
$m_s^*$	source mass flow	source cycle fouling, source pump malfunction (air channel fouling, air fan malfunction)
$m_r^*$	refrigerant mass flow	refrigerant line restriction, leakage
$c_{cmp}$ , $\lambda_{cmp}$	compressor: polytropic exponent mass flow supply efficiency	compressor malfunction
$k_{exp}$ , $\Delta T_{ov}$	expansion valve: opening superheating temp. diff.	expansion valve malfunction
$\varepsilon_{sw}$ , $\varepsilon_{ec}$	exergetic efficiency: source-to-water evaporator-to-condenser	used for fault detection (at least one of the faults above is present)

## 8.6 Gradual and Simultaneous Faults

Because of the physical nature of the parameters, gradual and even simultaneous faults can be recognized in a straightforward way. Gradual faults result in a varying size of the corresponding physical parameters. For example, an increasing fouling of the evaporator results in a decreasing value of the heat transfer parameter  $kA_e$  (cf. Table 16). Simultaneous faults are represented by changes in several physical parameters. For example, fouling of both the evaporator and the condenser results in lower values for both heat transfer parameters  $kA_e$  and  $kA_c$ . In Chapter 9.2 data from the IMRT test bench is used to test the diagnosis system with single, gradual, as well as simultaneous faults.

## 8.7 Online Fault Diagnosis

### 8.7.1 Diagnosis Steps

The online fault diagnosis system uses the reference parameter values from initialization and executes the following steps (cf. online software in Section 10.2):

- *Online data acquisition and preprocessing.* The online data is acquired for each data sequence. The data of the steady-state interval  $t_{ss} \dots t_{ss} + \Delta t_{ss}$  is stored.
- *Parameter identification.* The parameters  $\hat{\kappa} = [\hat{\kappa}_1, \hat{\kappa}_2, \dots]$  are identified by the method of Section 8.2.4 and compared to the reference values  $\kappa_{ref} = [\kappa_{1,ref}, \kappa_{2,ref}, \dots]$ . Afterwards, the parameter deviations  $\Delta \hat{\kappa}_i$  are scaled by the reference values  $\kappa_{i,ref}$  (eq. (93)). A value  $\Delta \hat{\kappa}_{i,sc} > 1$  indicates a positive parameter change and a value  $\Delta \hat{\kappa}_{i,sc} < 1$  indicates a negative parameter change.

$$\Delta \hat{\kappa}_{i,sc} = \frac{\Delta \hat{\kappa}_i}{\kappa_{i,ref}} = \frac{\hat{\kappa}_i - \kappa_{i,ref}}{\kappa_{i,ref}} = \frac{\hat{\kappa}_i}{\kappa_{i,ref}} - 1 \quad (93)$$

- *Statistical Evaluation.* In order to improve the reliability significantly, the identified parameters of several sequences are averaged (cf. Section 8.7.2).

### 8.7.2 Averaging

Each data sequence  $i_{seq}$  yields one parameter set  $\kappa(i_{seq})$ . Several data sequences are combined in one cycle  $i_{cyc}$  (Fig. 66).

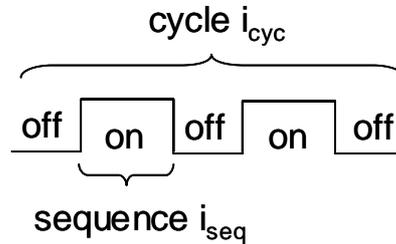


Fig. 66: Cycles and sequences

At the end of each cycle, the parameters are statistically evaluated by eq. (94), which calculates the mean values of the parameters for  $l$  sequences.

$$\bar{\kappa}(i_{cyc}) = \frac{1}{l} \sum_{s=1}^l \kappa(s) \quad (94)$$

### 8.7.3 Parameter Trend Charts

The cycle mean values of the parameters are plotted in trend charts (Fig. 67). These trend charts are informative for the progression of each parameter. With the knowledge of the actual parameters, a condition-oriented maintenance is possible.

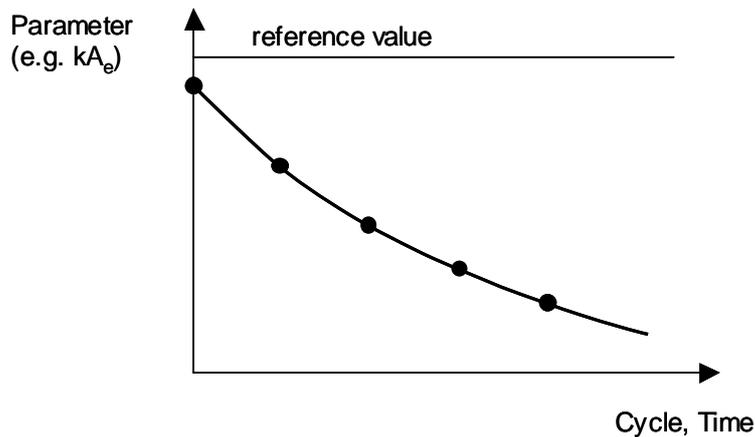


Fig. 67: Trend chart for one parameter over time, dots = mean values of each cycle.



## 9 A Fault Diagnosis System with Steady-State Parameter Identification – Applications

Data was acquired from a residential building at Barzheim (Switzerland) and from the test bench at the Measurement and Control Lab (IMRT). An overview is given in Table 1 on page 29 (column *HeatWatch*).

### 9.1 Data from Residential Building (Application I)

The diagnosis system has been tested with nominal data of one heating period (1998) from the residential building at Barzheim with an air-to-water heat pump (cf. Fig. 64 on page 142 and Fig. 84 on page 174). Seven temperature sensors, two pressure sensors, and one electrical power sensor are used (list (95)), whereas the electrical power is measured by a commercial low-cost energy counter.

$$\begin{aligned} & [T_{ov} \quad T_{hg} \quad T_{sc} \quad T_{s,i} \quad T_{s,o} \quad T_{w,i} \quad T_{w,o} \quad p_{lp} \quad p_{hp} \quad P_{HP}] \\ & = [T_3 \quad T_4 \quad T_6 \quad T_9 \quad T_7 \quad T_{10} \quad T_8 \quad p_{lp} \quad p_{hp} \quad P_{HP}] \end{aligned} \quad (95)$$

The durations of the ‘on’ data sequences typically vary between 60 and 90 minutes (between the defrosting sequences). The duration of the steady-state interval  $\Delta t_{ss}$  is chosen to be 20 minutes, whereas the steady-state starting time  $t_{ss}$  is automatically detected (cf. Section 8.3). With a sampling time of 10 seconds, 120 samples are averaged during the steady-state interval.

Within each day of the heating period, the parameters are identified for all data sequences (mode on). The relative standard deviations of the eleven parameters are shown in Fig. 68 for each day. They are below or around 10% for seven parameters ( $kA_e$ ,  $m_w^*$ ,  $m_r^*$ ,  $c_{cmp}$ ,  $\lambda_{cmp}$ ,  $\Delta T_{ov}$ ,  $k_{exp}$ ), below 20% for three parameters ( $kA_{he,ov}$ ,  $kA_c$ ,  $m_a^*$ ), while only one parameter ( $kA_{he,e}$ ) reaches the 20% mark.

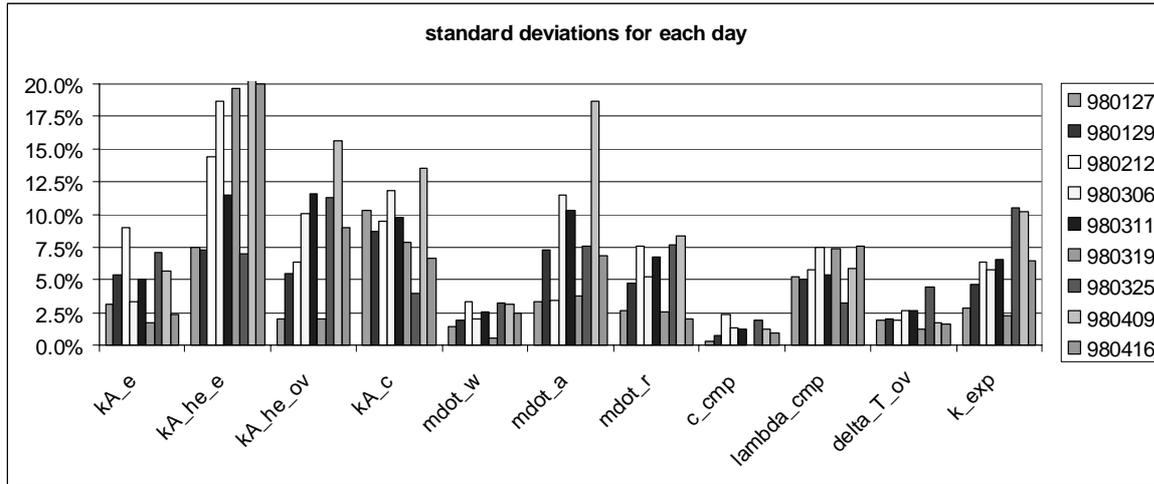


Fig. 68: Relative standard deviations of the identified parameters for each day.

The relative standard deviations of five characteristics  $COP_{HP}$ ,  $COP_{Carnot,aw}$ ,  $COP_{Carnot,ec}$ ,  $\epsilon_{aw}$ ,  $\epsilon_{ec}$  are shown in Fig. 69.

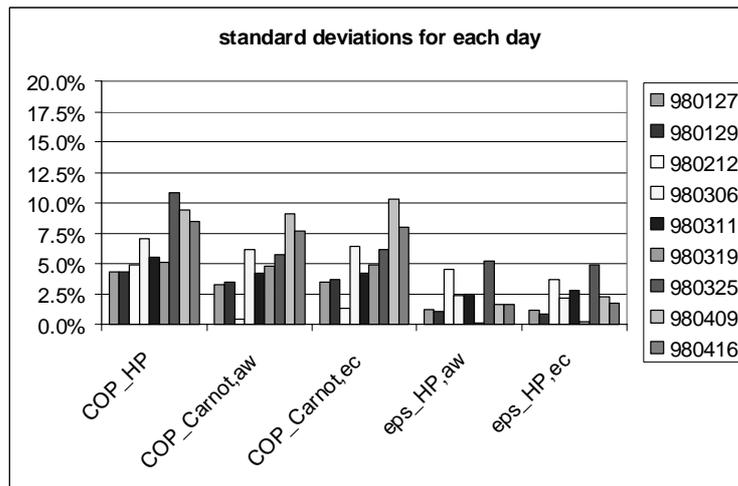


Fig. 69: Relative standard deviations of the coefficient of performance ( $COP_{HP}$ ), the Carnot coefficient ( $COP_{Carnot}$ ), and the quality grade ( $\epsilon_{HP}$ ).  $aw$  = air-to-water,  $ec$  = evaporator-to-condenser

Ten additional characteristics are calculated, as well:

- the energy consumption  $E_{HP}$  and the energy output  $Q_{HP}$
- the energetic efficiency  $PF_{HP} = Q_{HP}/E_{HP}$
- the energy consumption  $E_{HP,defrost}$  during the defrosting cycle
- the operating hours  $\Delta t_{on}$  and the defrosting hours  $\Delta t_{defrost}$
- the output heat flow  $Q_{HP}^*$ , the source heat flow  $Q_s^*$  and the heat flows  $Q_{he,e}^*$ ,  $Q_{he,ov}^*$  in the internal heat exchanger.

The energetic quantities are integrated over one data sequence, whereas the heat flows are calculated for steady-state.

The parameter trends within the days are shown in the following figures. The evaporator heat transfer parameter  $kA_e$  clearly shows the icing effect (Fig. 70, scaled to 0..100%). For those days in winter with a low air temperature and a high air humidity (98-01-27, 98-01-29, ...), the parameter  $kA_e$  is lower than for those days in spring with a higher air temperature (98-04-09, ...). The parameters are identified before the starting of the defrosting cycles. As an option, the parameter  $kA_e$  could be used for detecting ice and starting the defrosting cycle (cf. Section 14). Whereas the variations between the different days are high, the variations within each day are low (which is shown in Fig. 68 also). Fig. 71 shows that the compressor is working with a rather constant efficiency.

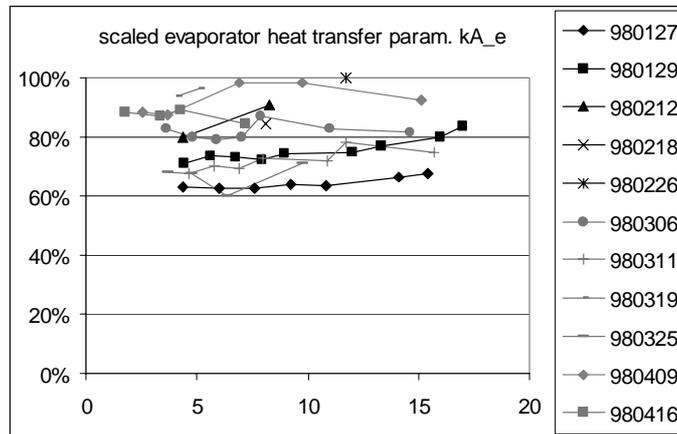


Fig. 70: Scaled evaporator heat transfer parameter  $kA_e$ , trends for each day.

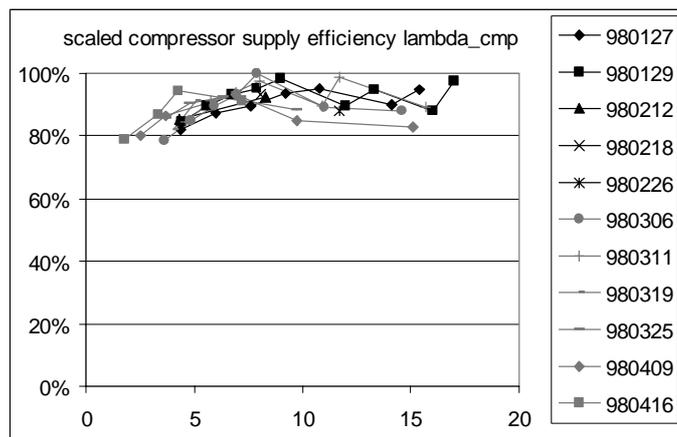


Fig. 71: Scaled compressor mass flow supply efficiency  $\lambda_{cmp}$ , trends for each day.

The variations of the refrigerant mass flow  $m_r^*$  is higher than the variation of the water mass flow  $m_w^*$  through the circulating pump (Fig. 72 and Fig. 73), since  $m_r^*$  is controlled by the expansion valve.

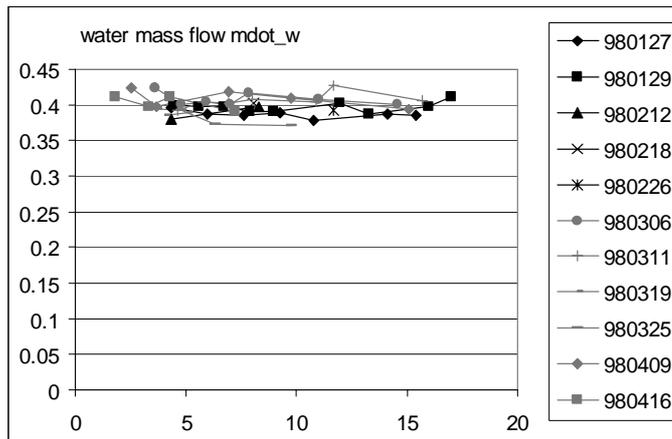


Fig. 72: Water mass flow  $m_w^*$  in kg/s, trends for each day.

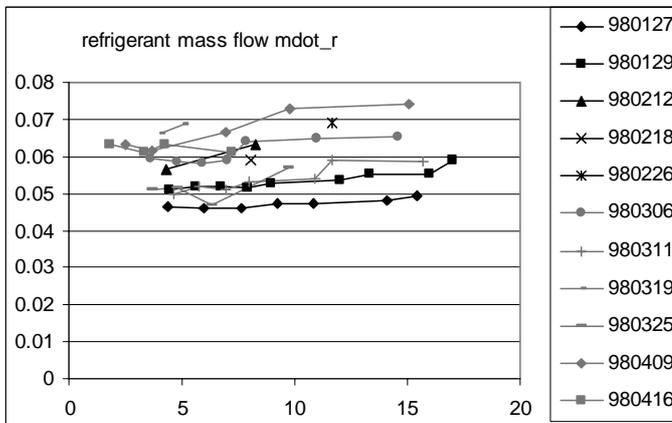


Fig. 73: Refrigerant mass flow  $m_r^*$  in kg/s, trends for each day.

Comparing the coefficient of performance  $COP_{HP}$  (Fig. 74) with the (air-to-water) exergetic efficiencies  $\epsilon_{aw}$  (Fig. 75), the lower variations of  $\epsilon_{aw}$  are obvious. Thus, the exergetic efficiency can be used for fault detection (decision, whether or not any fault is present, [Kuijk 96]).

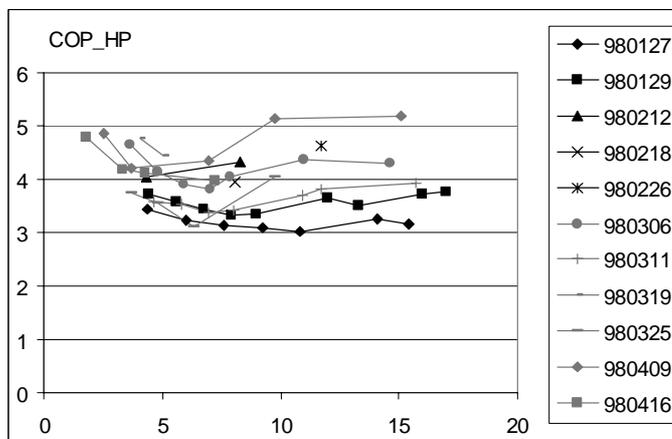


Fig. 74: Coefficient of performance  $COP_{HP}$ , trends for each day.

9.1 Data from Residential Building (Application I)

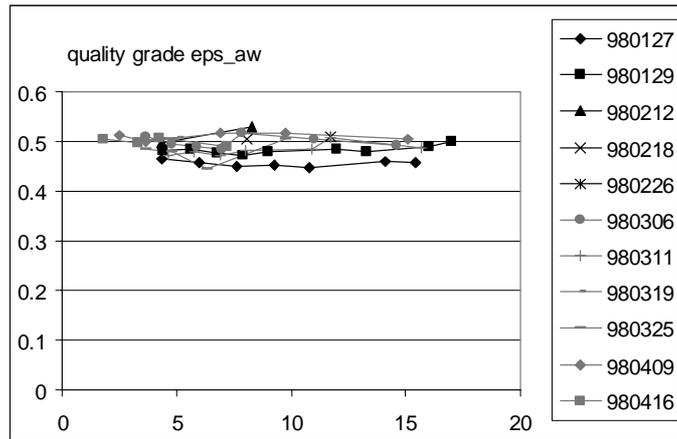


Fig. 75: Exergetic performance  $\epsilon_{aw}$  (air-to-water): trends for each day.



## 9.2 Data from IMRT Test Bench (Application IV)

The diagnosis system of Chapter 8 was tested with the data of the nominal case as well as with several fault cases using the IMRT test bench. The software of Chapter 10 was used to initialize the diagnosis system and to test the online diagnosis. The results are significantly improved by averaging over several data sequences.

### 9.2.1 Configuration of the Test Heat Pump

The modified brine-to-water heat pump of the IMRT test bench introduced in Section 6.3.1 is also used here. But here other sensors are required, which are shown in Fig. 76.

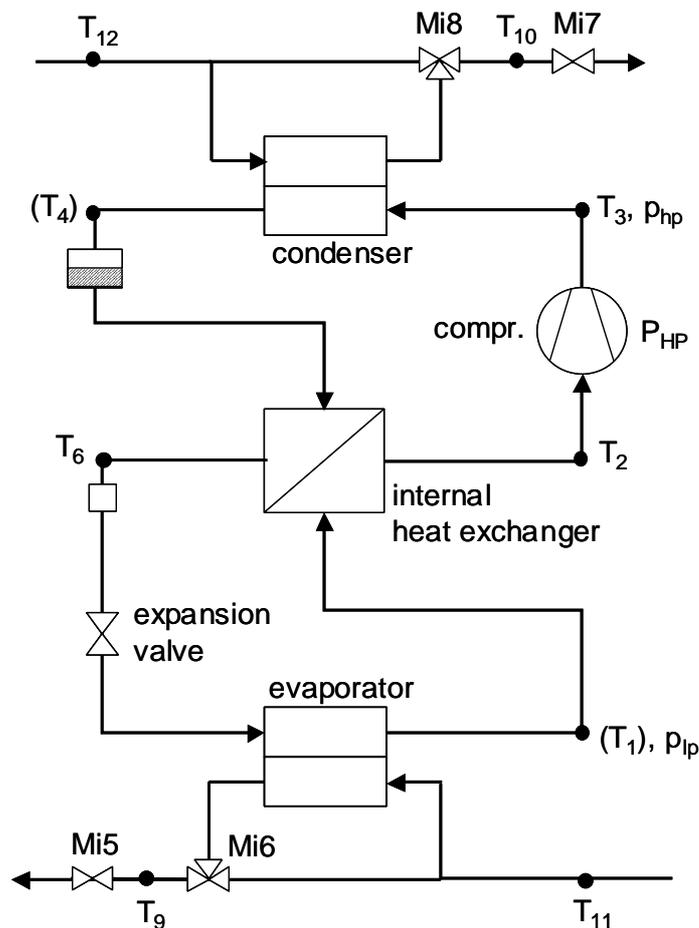


Fig. 76: The modified brine-to-water heat pump, with additional valves (Mi5...Mi8), temperature sensors ( $T_i$ ), pressure sensors ( $p_i$ ) and an electrical power sensor ( $P_{HP}$ ).

Nine temperature sensors, two pressure sensors, and one electrical power sensor are used (eq. (96)), whereas the electrical power is measured by a commercial low-cost energy counter. The additional sensors  $T_1$  and  $T_4$  replace the calculated temperatures  $T_d(p_{lp})$  and  $T_b(p_{hp})$  (cf. Section 8.2.2). Without a significant decrease in the diagnosis quality, they can be eliminated, which reduces the number of temperature sensors to seven, as described in Section 9.1.

$$\begin{aligned} & \left[ T_{ov} \quad T_{hg} \quad T_{sc} \quad T_{s,i} \quad T_{s,o} \quad T_{w,i} \quad T_{w,o} \quad p_{lp} \quad p_{hp} \quad P_{HP} \quad T_d(p_{lp}) \quad T_b(p_{hp}) \right] \\ & = \left[ T_2 \quad T_3 \quad T_6 \quad T_{11} \quad T_9 \quad T_{12} \quad T_{10} \quad p_{lp} \quad p_{hp} \quad P_{HP} \quad T_1 \quad T_4 \right] \end{aligned} \quad (96)$$

The durations of the data sequences typically vary between 10 and 30 minutes. Here, a fixed steady-state starting time at  $t_{ss} = 9$  minutes and a steady-state interval duration of  $\Delta t_{ss} = 1$  minute are selected (cf. Section 8.3). With a sampling time of 1 second, 60 samples are averaged during the steady-state interval.

The five fault cases are listed in Table 10 on page 116. A total of eleven parameters are identified (cf. Table 16 on page 149). Table 17 shows the four principal parameters, which are used for interpreting the five fault cases (including the nominal case).

Table 17: Principally influenced parameters and corresponding faults in the IMRT test bench.

changed parameter	parameter description	fault interpretation for IMRT test bench
-	-	fault 1: no fault
$m_s^*$	source mass flow	fault 2: valve Mi5
$kA_e$	evaporator heat transfer	fault 3: valve Mi6
$m_w^*$	water mass flow	fault 4: valve Mi7
$kA_c$	condenser heat transfer	fault 5: valve Mi8

### 9.2.2 Diagnosis Results

The diagnosis system was tested for the data of the faults above. During initialization, the reference parameters were calculated by averaging them over five sequences of one nominal data cycle. During online diagnosis, the deviations between the actual parameters and the reference parameters are calculated. If one parameter has a significantly higher deviation than all other

eleven parameters, the corresponding fault is diagnosed. Thus, the diagnosis quality is checked by counting the number of wrong diagnosis results.

Table 18 shows the rate of wrong diagnosis results for different validation data sets with averaging (cf. Section 8.7.2). For each cycle, the identified parameters of five data sequences are averaged. In the left column, the results for single faults are listed for different fault sizes. For the sizes 20% and 30%, the number of wrong diagnosis results is zero with the data used. Small faults (size 10%) are harder to distinguish from the nominal case because the parameter deviations are small, whereas large faults (size 50%) produce large deviations in all parameters. Therefore the localization of the fault is more difficult. But in general, the system is able to diagnose gradual faults very well.

*Table 18: Rate of wrong diagnosis results with single and simultaneous faults of different sizes.*

fault size	“online” validation averaging over 5 sequences	
	single faults	single faults + simultaneous faults
10%	1/31 = 3.2%	
20%	0/24 = 0.0%	1/44 = 2.3%
30%	0/31 = 0.0%	0/52 = 0.0%
50%	2/19 = 10.5%	

The fault diagnosis system was also checked with simultaneous faults, which is shown in the right-hand column of Table 11. Data of nine fault cases, including five single faults as well as four combinations of two simultaneous faults were acquired (cf. list on page 125). For the faults of size 20% and 30%, the diagnosis quality is high.

For fault detection, the exergetic efficiencies  $\varepsilon_{sw}$  and  $\varepsilon_{ec}$  are used. An even better indicator is defined by  $\Delta\kappa_{max}$ , which is the maximum of all parameter deviations. If the maximum deviation  $\Delta\kappa_{max}$  is high, a fault must be present. A false alarm occurs if  $\Delta\kappa_{max}$  is high, even if no fault is present. With the validation data of the nominal case, no false alarms have occurred.



## 10 Software HeatWatch

The diagnosis system of Chapter 8 is realized by a software tool, written in MATLAB<sup>®</sup>. The software tool includes the initialization procedure as well as the online operation, and provides a graphical user interface (GUI). The software tool has been adapted to the data of the IMRT test bench (cf. Section 9.2 and Chapter 12).

At startup, the software tool displays the main panel (Fig. 77), which contains the scheme of the heat pump. First, the classification system has to be initialized, as described in Section 10.1. The panel then serves to display the classification results for online diagnosis (cf. Section 10.2). All parameters are described in Table 16 on page 149, whereas in the software the following symbols are used:

$$kA_e = kA_e$$

$$kA_{he_e} = kA_{he,e}$$

$$kA_{he_{ov}} = kA_{he,ov}$$

$$kA_c = kA_c$$

$$\dot{m}_w = \dot{m}_w^*$$

$$\dot{m}_s = \dot{m}_s^*$$

$$\dot{m}_r = \dot{m}_r^*$$

$$c_{cmp} = c_{cmp}$$

$$\lambda_{cmp} = \lambda_{cmp}$$

$$k_{exp} = k_{exp}$$

$$dT_{ov} = \Delta T_{ov}$$

$$(COP_{HP} = COP_{HP})$$

$$\epsilon_{sw} = \epsilon_{sw}$$

$$\epsilon_{ec} = \epsilon_{ec}$$

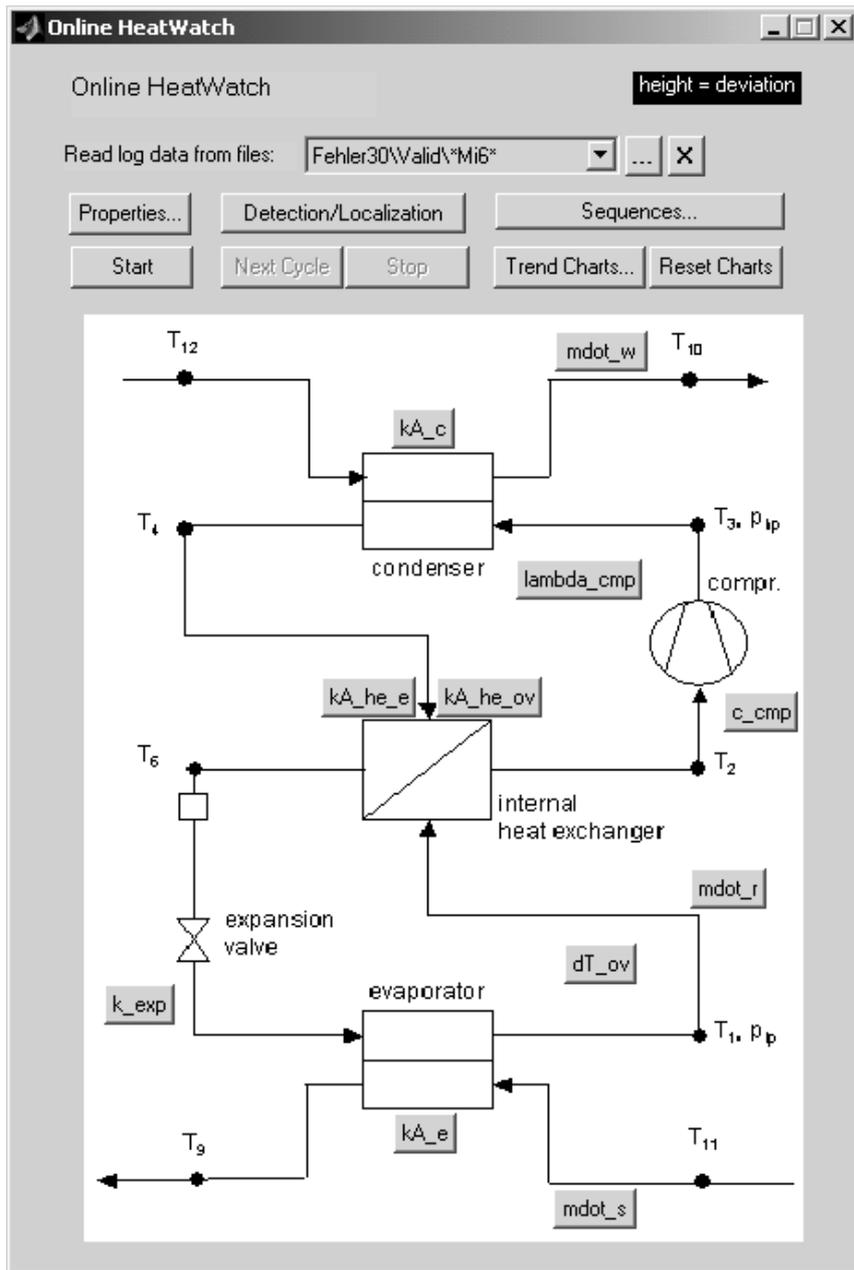


Fig. 77: Main panel.

## 10.1 Initialization and Properties

By pressing the button ‘properties’ on the main panel, the properties panel of Fig. 78 is activated. The diagnosis system is initialized by pressing the button ‘Set parameters of current cycle as reference’ on this panel. Therefore, the current parameters are stored in the reference parameter file. During operation, the actual parameters are compared to the reference values loaded from this file. For online diagnosis, the number of averaged data sequences per cycle is defined (cf. Section 8.7.2).

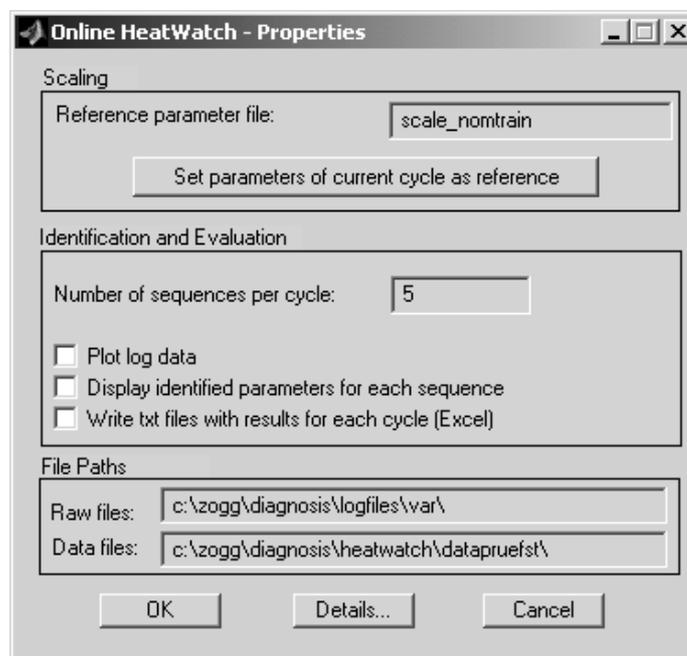


Fig. 78: Properties and initialization of the diagnosis system.

Optionally, more detailed properties may be defined on the panel in Fig. 79 (activated by pressing the button ‘details’ in the panel above). In the first paragraph, a minimum duration of the data sequences is defined. In the next paragraph, either an automatic steady-state detection is selected (by selecting the corresponding check box) or a fixed steady-state time is chosen (cf. Section 8.3). For an automatic detection, the detection signal  $x$  is chosen and the derivative limit  $dx_{limit}$  is defined. Else, the steady-state end time ( $t_{ss} + \Delta t_{ss}$ ) and the steady-state interval duration  $\Delta t_{ss}$  is chosen.

Online HeatWatch - Properties - Details

Log Data

Minimal data length [min]: 1

Steady State Detection

Detection signal number: 3

Derivative limit for detection signal: 0.00035

Steady state end time [min]: 10

Steady state interval length [min]: 1

Parameter display scalings: deviations for full scale

kA_e	1	c_cmp	1
kA_he_e	1	lambda_cmp	1
kA_he_ov	1.6	k_exp	1
kA_c	1	dT_ov	1
mdot_w	0.5	COP_HP	1
mdot_s	0.8	eps_sw	0.4
mdot_r	1	eps_ec	0.3

OK Cancel

Fig. 79: Detailed properties for steady-state detection and visualization.

For the online diagnosis described in Section 10.2, the parameters are displayed with bars, the heights of which indicate the deviations of the parameters from their reference values. In the last paragraph of Fig. 79, the scalings of these parameter displays are defined, which are used to weight the parameter deviations. By default, all scalings are set to 1. During the first diagnosis cycles, these scalings may be adjusted, as done here.

## 10.2 Online Diagnosis

After initialization, the diagnosis software is applied to the online data from operation (validation data). Therefore, the online fault diagnosis steps of Section 8.7.1 are executed repetitively for each data sequence.

### 10.2.1 Sequential Parameter Identification

For the following evaluations, the same data sets are used as those listed in Section 9.2.2. In Fig. 80, validation data of the fault case 3 (evaporator fouling, cf. Table 10 on page 116) is evaluated.

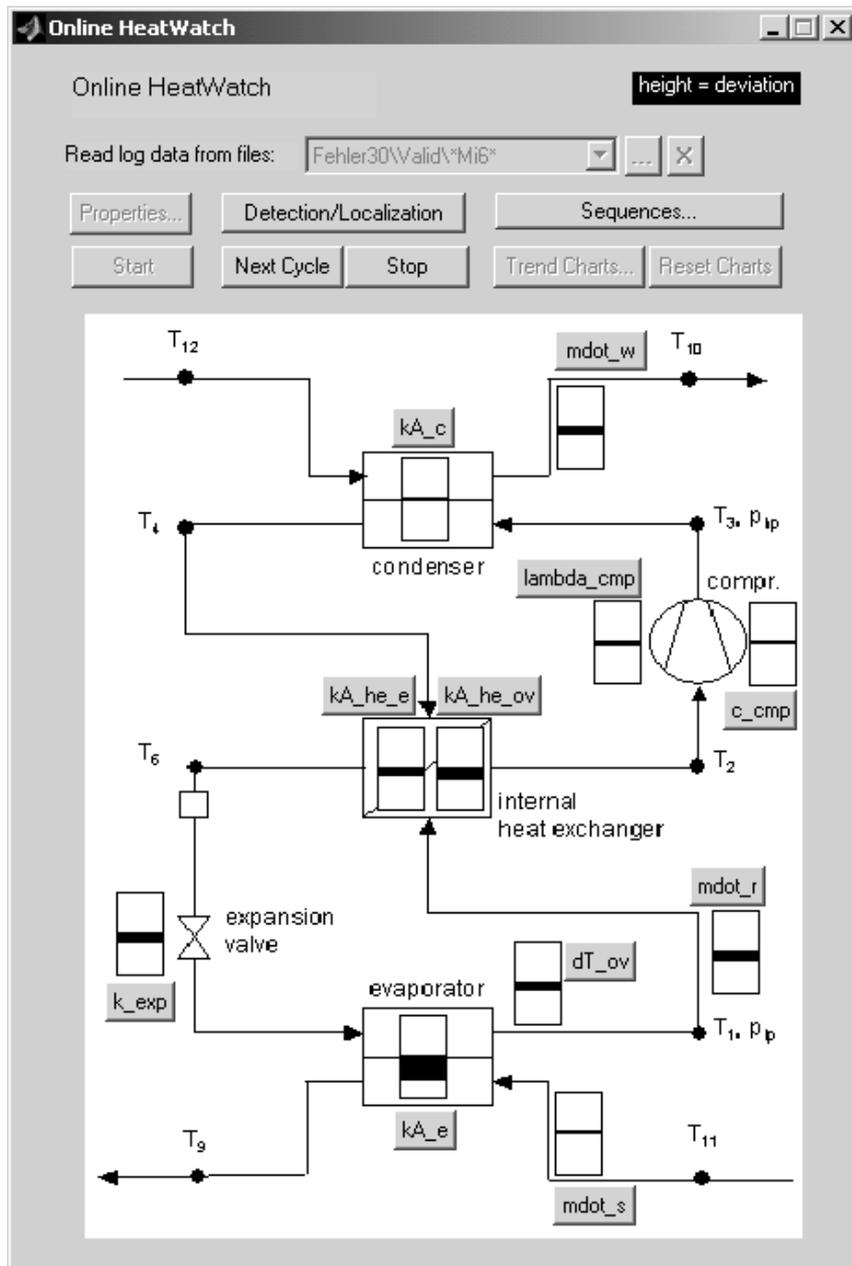


Fig. 80: Identification results with validation data for fault 3 (evaporator fouling), mean values of the parameters for one cycle. Large bar height for parameter  $kA_e$  (evaporator heat transfer) = large deviation from reference value. Small bar heights for other parameters = small deviations.

Here, the identified parameters are averaged over five sequences for one sample cycle. The parameter deviations are visualized by the heights of the bars, which also indicate the directions of the deviations. Fig. 80 shows that fault 3 (evaporator fouling) results in the highest deviation for the parameter  $kA_e$  (evaporator heat transfer). The reduction of the evaporator heat transfer parameter also results in deviations of other parameters, since all parameters are physically coupled. In this case, the increasing filling grade of the evaporator also reduces the superheating temperature difference  $\Delta T_{ov}$ , the superheating heat transfer parameter  $kA_{he,ov}$  (because of a changed heat transfer area  $A$ ), the expansion valve opening  $k_{exp}$  (as a reaction of the P-controller to a change of  $\Delta T_{ov}$ ), and the refrigerant mass flow  $m_r^*$  (as a result of the changed opening  $k_{exp}$ ). Nevertheless, the fault can be clearly localized from the highest deviation in the parameter  $kA_e$ .

The fault diagnosis system directly locates the parameter deviations. Nevertheless, a two-step display with a fault detection view and a fault localization view is useful in practice. By pressing the button “Detection/Localization”, the view may be changed (Fig. 81). In the detection view, the ‘state’ of the machine indicates whether there is any parameter change (dark indicator) or not (white indicator). The ‘state’ is given by the maximum of the parameter deviations for the current cycle (according to the mean values of Fig. 80, the maximum is given by the deviation of the parameter  $kA_e$ ). Moreover, the exergetic efficiencies  $\varepsilon_{sw}$  (source-to-water) and  $\varepsilon_{ec}$  (evaporator-to-condenser) are well suited for detecting deviations in the performance caused by any faults. Here the negative deviation in the first exergetic efficiency is very high.

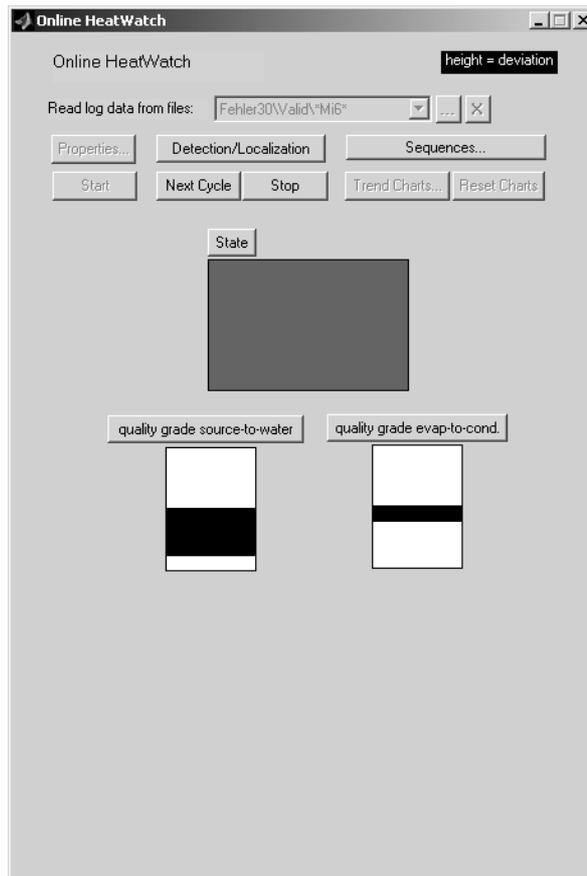


Fig. 81: Identification results with validation data for fault 3 (evaporator fouling), fault detection view. 'State' is dark shaded = a fault is present, bar height of source-to-water exergetic efficiency is high = large deviation

Fig. 82 shows the identified parameters, the coefficient of performance  $COP_{HP}$ , as well as the exergetic efficiencies  $\varepsilon_{sw}$  and  $\varepsilon_{ec}$  for each data sequence of the cycle above. For each parameter, the columns correspond to the five sequences. In the additional columns, the mean values of the parameters are displayed. Again, the height of the bars indicates the size and the direction of any deviation. All sequences have high deviations in the parameter  $kA_e$  (evaporator heat transfer) and the characteristic value  $\varepsilon_{sw}$  (exergetic efficiency, source-to-water). The diagnosis is correct for all sequences as well as for their mean values.



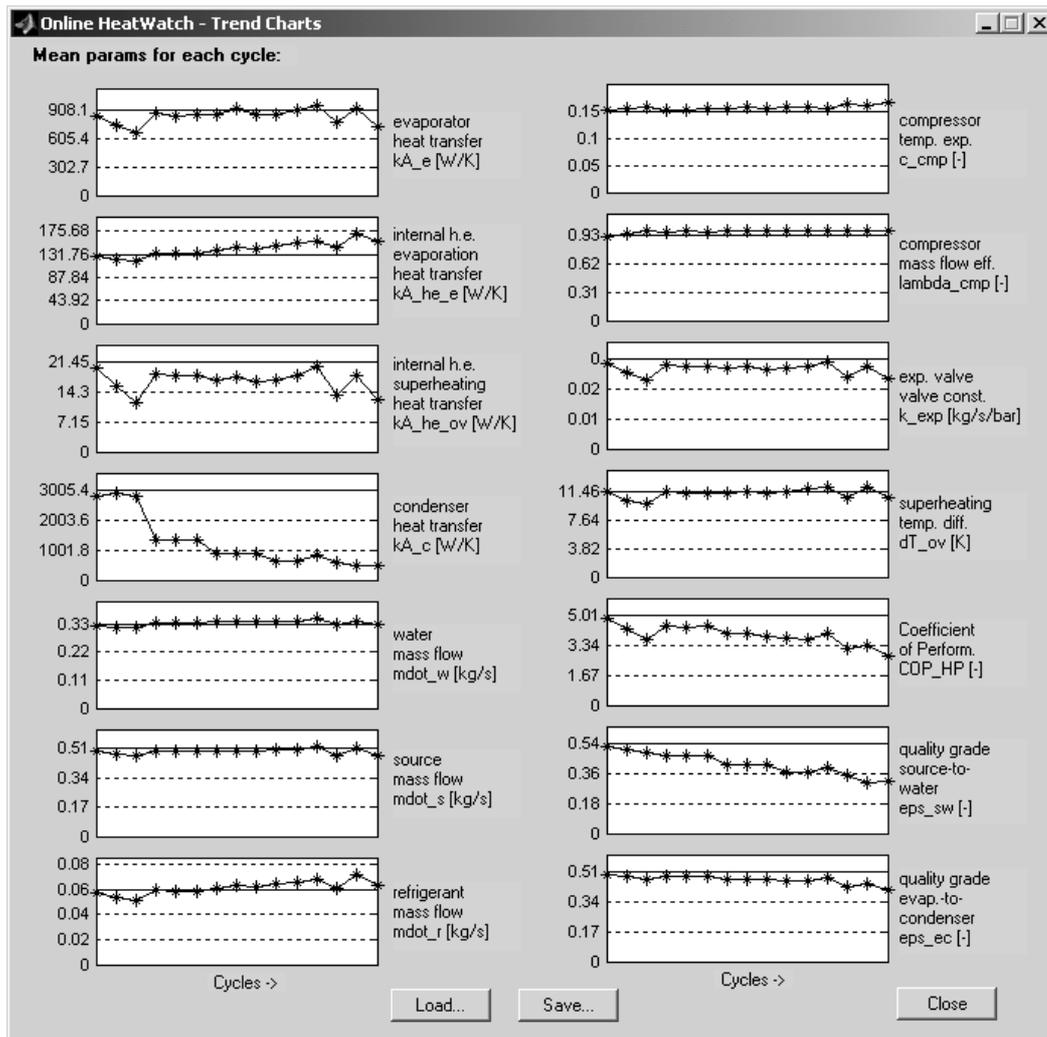


Fig. 83: Trend charts for validation data of fault 5 (condenser fouling) with an increasing fault size (decreasing condenser heat transfer parameter  $kA_c$ ). The reference values are plotted as horizontal lines. Each point (\*) corresponds to one cycle mean value.



## 11 Simulation Model

The simulation model has been used to generate data for training and testing the diagnosis system presented in Section 6.2. Thus, it has been used as a ‘virtual’ test bench. But the fault diagnosis systems are *not* based on this model, since it is much too complex for online use. In this chapter, the physical equations of the detailed dynamic model are presented. It has been developed for simulating a commercial air-to-water heat pump from SATAG<sup>®</sup> Thermotechnik AG and it is validated for the nominal case by using measured data from the residential building at Barzheim. The model is then used to simulate several fault cases by changing the model parameters. The model code has been programmed in C/C++ and then compiled for speed-up.

### 11.1 Modeling the Heat Pump Refrigerant Cycle

The simulation model represents the heat pump shown in Fig. 84. The data sheet of the heat pump is given in Appendix C, Section ‘SATAG Natura AW110’. The heat source is represented by the ambient air and the heat sink is represented by the water cycle of the residential building. This type of heat pump contains an internal heat exchanger and a hot gas defrosting cycle. The internal heat exchanger transfers the heat flow from the subcooling process to the superheating process and improves the performance. At low air temperature and high air humidity, the icing decreases the heat transfer rate of the evaporator. Therefore the defrosting cycle must be activated periodically by feeding the evaporator with hot gas.

The refrigerant cycle process has been divided into several sub-processes, for which the models are developed in the next Sections. The model is designed for all three modes of the heat pump (on, off, defrosting). The refrigerant migration from the high-pressure side to the low-pressure side during the shut-down sequences as well as the activation of the hot gas bypass during the defrosting sequences are modelled. The migration process is also described in [Gubser 99]. For the presentation in this work, the equations of the program code are simplified by neglecting some unimportant terms and special cases.

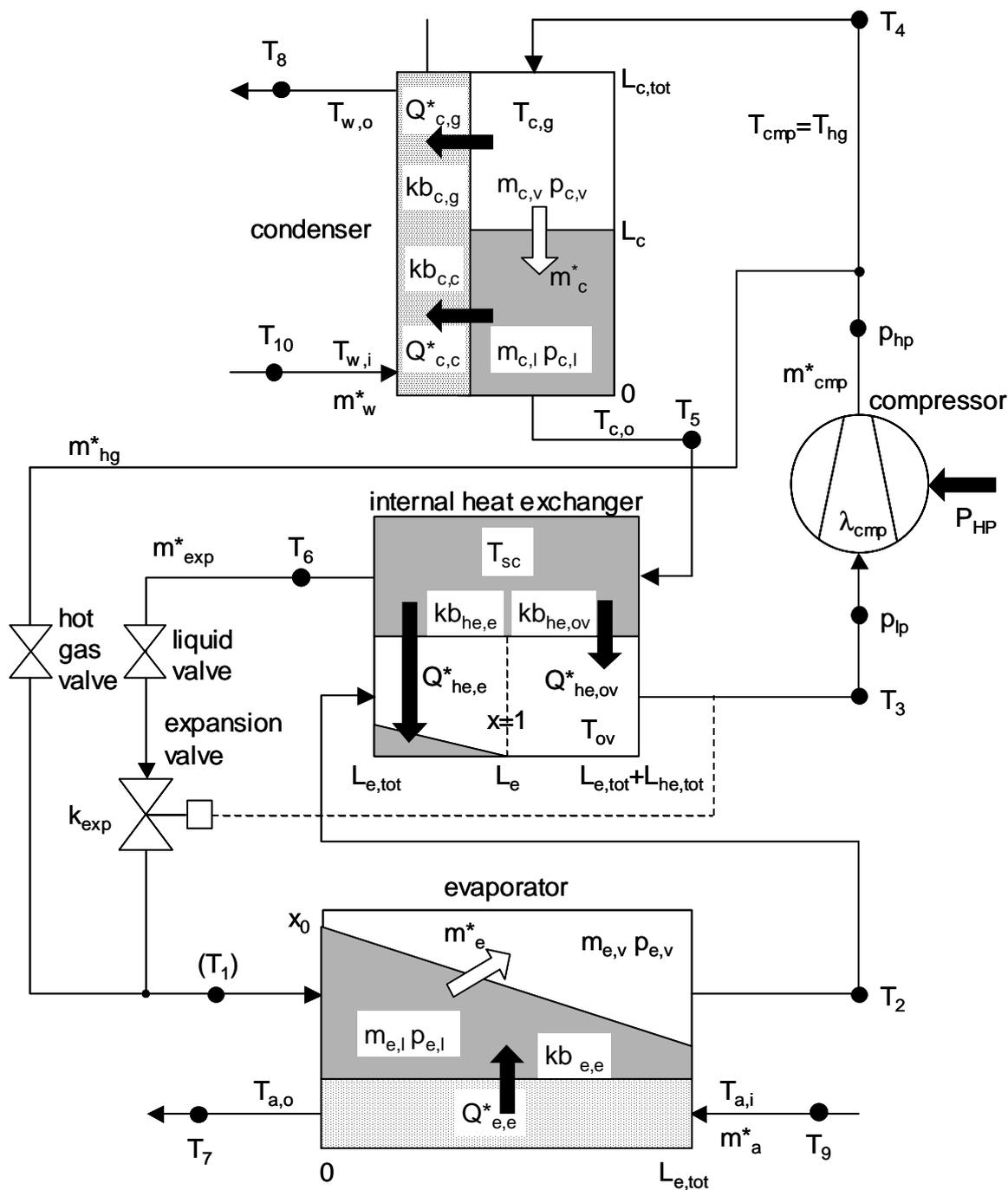


Fig. 84: Refrigerant cycle of the air-to-water heat pump with an internal heat exchanger.

According to the mode  $u_{mode}$ , the control inputs for the compressor  $u_{cmp}$ , the liquid valve  $u_{lv}$ , the hot-gas valve  $u_{hgv}$ , the air fan  $u_{fan}$ , and the water circulation pump  $u_{pump}$  are set to appropriate values (eq. (97)). During the shut-down sequence ( $u_{mode} = 0$ ), the circulating pump keeps running ( $u_{pump} = 1$ ), because the heat pump is controlled by measuring the actual return flow water temperature. The liquid valve is closed ( $u_{lv}=0$ ) to minimize refrigerant migration.

For the defrosting mode ( $u_{mode}=2$ ), the hot-gas valve is opened ( $u_{hgv}=1$ ) and the fan and the circulating pump are stopped ( $u_{fan}=0$ ,  $u_{pump}=0$ ) to prevent heat losses.

$$\begin{bmatrix} u_{cmp} \\ u_{lv} \\ u_{hgv} \\ u_{fan} \\ u_{pump} \end{bmatrix} = \begin{cases} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} & (u_{mode} = 0 = off) \\ \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} & (u_{mode} = 1 = on) \\ \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} & (u_{mode} = 2 = defrosting) \end{cases} \quad (97)$$

### 11.1.1 Evaporation

The liquid refrigerant evaporates while it is flowing through the evaporator tubes. At the inlet the vapour mass ratio  $x=x_{e,i}$  is defined by (98).

$$x_{e,i} = \frac{m_{v,e,i}}{m_{e,i}} \quad (98)$$

At the position  $L_e$  there is no liquid left ( $x=1$ ), thus  $L_e$  is called the “evaporation length”. Generally,  $L_e$  is time-varying, which is considered by the model [Gruhle 87].

The balance for the liquid mass  $m_{e,l}$  is given by eq. (99), with  $m_{exp,l}^*$  and  $m_{hg,c}^*$  being the liquid part of the input mass flows from the expansion valve and the defrosting cycle (cf. eqs. (153) and (158)), and  $m_e^*$  representing the evaporating mass flow.

$$\frac{dm_{e,l}}{dt} = m_{exp,l}^* + m_{hg,c}^* - m_e^* \quad (99)$$

By using the derivative  $dm_{e,l}/dt$  from eq. (99), the differential equation for  $L_e$  is defined (eq. (100)) with the liquid density  $\rho'(p_{e,l})$ , the evaporator cross-sectional area  $A_e$ , and the mean cross-sectional area ratio  $\beta_e$  of the liquid phase. Regarding an evaporator with bundles of pipes,  $A_e$  is the sum of the areas over all the parallel pipes.

$$\frac{dL_e}{dt} = \frac{dm_{e,l}}{dt} \cdot \frac{1}{\rho'(p_{e,l}) \cdot A_e \cdot \beta_e} \quad (100)$$

Usually the evaporation process is not finished at the outlet of the evaporator, but extends to the internal heat exchanger. Thus,  $L_e$  has to be divided into two parts,  $L_{e,e}$  for the evaporator and  $L_{he,e}$  for the internal heat exchanger. After the evaporation process, there is the superheating process, which usually takes place in the internal heat exchanger ( $L_e >$  total length of the evaporator  $L_{e,tot}$ ). During certain transients it is possible that the evaporator is almost empty. Then the superheating process takes place in the evaporator already ( $L_e < L_{e,tot}$ ) and the “superheating length”  $L_{ov}$  has to be divided into two parts,  $L_{e,ov}$  as well as  $L_{he,ov}$ . The cases above are represented by the second and third cases of eq. (101). The first case in eq. (101) stands for an “overfilled” evaporator as well as an “overfilled” internal heat exchanger ( $L_e > L_{e,tot} + L_{he,tot}$ ). In this case  $L_e$  is defined virtually as being outside of the geometric limitations. If the volume of the liquid phase  $V_{e,l}$  (c.f. eq. (112)) reaches the maximum volume of the evaporator and of the internal heat exchanger, then  $L_e$  will reach the limit  $L_{e,limit}$  and will not increase anymore.

$$\begin{pmatrix} L_{e,e} \\ L_{e,ov} \\ L_{he,e} \\ L_{he,ov} \end{pmatrix} = \begin{cases} \begin{pmatrix} L_{e,tot} \\ 0 \\ L_{he,tot} \\ 0 \end{pmatrix} & (L_{e,tot} + L_{he,tot} \leq L_e \leq L_{e,limit}) \\ \begin{pmatrix} L_{e,tot} \\ 0 \\ L_e - L_{e,tot} \\ L_{e,tot} + L_{he,tot} - L_e \end{pmatrix} & (L_{e,tot} \leq L_e < L_{e,tot} + L_{he,tot}) \\ \begin{pmatrix} L_e \\ L_{e,tot} - L_e \\ 0 \\ L_{he,tot} \end{pmatrix} & (0 \leq L_e < L_{e,tot}) \end{cases} \quad (101)$$

The balance for the vapour mass  $m_{e,v}$  is given by eq. (102), with  $m_{exp,g}^*$  and  $m_{hg,g}^*$  being the gaseous parts of the input mass flow from the expansion

valve and the defrosting cycle (cf. eqs. (154) and (158)), with  $m_e^*$  representing the evaporating mass flow and  $m_{cmp}^*$  representing the output mass flow to the compressor.

$$\frac{dm_{e,v}}{dt} = m_{exp,g}^* + m_{hg,g}^* + m_e^* - m_{cmp}^* \quad (102)$$

The energy balance for the liquid phase is expressed by eq. (103), with the time derivative of the pressure  $dp_{e,l}/dt$  and the partial derivative of the energy  $du'_e/dp_{e,l}$ , subject to the evaporation pressure  $p_{e,l}$  in the liquid phase.

$$\frac{dp_{e,l}}{dt} \frac{du'_e}{dp_{e,l}} m_{e,l} + u'_e \frac{dm_{e,l}}{dt} = [Q_{e,l}^* + Q_{e,e}^* + Q_{he,e}^*] \quad (103)$$

The convective heat flow  $Q_{e,l}^*$  is a sum of several heat flows, depending on the operation mode of the heat pump. When the heat pump is switched on during the normal operating mode, eq. (104) is valid, with the liquid enthalpy  $h_{exp,l}$  at the input (eq. (159)) and the saturated vapour enthalpy  $h''(p_{e,l})$  at the output.

$$Q_{e,l,1}^* = m_{exp,l}^* \cdot h_{exp,l} - u_{cmp} \cdot m_e^* \cdot h''(p_{e,l}) \quad (104)$$

For the defrosting mode an additional hot gas heat flow is introduced (eq. (105)). It depends on the hot gas enthalpy  $h_v$ , which is directly coming from the compressor outlet at the high temperature  $T_{cmp,hp}$  and the high pressure  $p_{c,v}$ . It is assumed that the part  $m_{hg,c}^*$  of the hot gas mass flow is condensed and that it transfers its high enthalpy to the evaporator.

$$Q_{e,l,2}^* = m_{hg,c}^* \cdot h_v(T_{cmp,hp}, p_{c,v}) \quad (105)$$

When the compressor is switched off ( $u_{cmp} = 0$ ), the high and low pressure levels will be equalized by the refrigerant migration process [Gubser 99]. During this process, the refrigerant condenses in the evaporator. Therefore the first term of eq. (106) considers a negative mass flow  $m_e^* < 0$ . The vapour enthalpy  $h_v$  is an input from the superheating process with the superheating temperature  $T_{ov}$  and the vapour pressure  $p_{e,v}$ .

$$Q_{e,l,0}^* = -(1 - u_{cmp}) \cdot m_e^* \cdot h_v(T_{ov}, p_{e,v}) \quad (106)$$

The total heat flow  $Q_{e,l}^*$  is the sum of these heat flows (eq. (107)).

$$Q_{e,l}^* = Q_{e,l,0}^* + Q_{e,l,1}^* + Q_{e,l,2}^* \quad (107)$$

The heat flow  $Q_{e,e}^*$  in eq. (103), which describes the heat transfer in the evaporator, is driven by the difference between the mean air temperature  $\bar{T}_a$  and the mean liquid refrigerant temperature  $\bar{T}_{e,l}$  (eq. (108)).

$$Q_{e,e}^* = kb_{e,e} \cdot L_{e,e} \cdot (\bar{T}_a - \bar{T}_{e,l}) \quad (108)$$

The heat flow  $Q_{he,e}^*$  in eq. (103) between the subcooled refrigerant and the evaporated refrigerant in the internal heat exchanger is defined by eq. (109).

$$Q_{he,e}^* = kb_{he,e} \cdot L_{he,e} \cdot (\bar{T}_{sc} - \bar{T}_{e,l}) \quad (109)$$

All the heat transfer parameters  $kb_i$  of this model are defined by eq. (110) with  $k_i$  for the heat transfer coefficient (in  $W/m^2K$ ),  $b_i$  for the width of the considered heat transfer element,  $L_i$  for its length, and  $A_i$  for the heat transfer area. In the case of a heat exchanger with bundles of tubes (evaporator),  $b_i$  is equal to the circumference of one tube, multiplied by the number of parallel tubes. In the case of a heat exchanger consisting of plates (condenser, cf. Section 11.1.5),  $b_i$  is equal to the width of one plate, multiplied by the number of parallel plates.

$$kb_i = k_i \cdot b_i \quad kA_i = k_i \cdot A_i = k_i \cdot b_i \cdot L_i \quad (110)$$

Whereas the pressure of the liquid phase  $p_{e,l}$  is described by the differential equation (103), the pressure of the vapour phase  $p_{e,v}$  is calculated from the gas equation (111) with the temperature  $T_{ov}$  and the specific volume  $v_{e,v}$ . For this equation, the temperature in the vapour phase of the evaporation process is considered to be equal to the superheating temperature  $T_{ov}$  in the superheating process (cf. next Section), which is a simplification.

$$p_{e,v} = p_{e,v}(T_{ov}, v_{e,v}) \quad v_{e,v} = \frac{V_{e,v}}{m_{e,v}} \quad (111)$$

$$V_{e,v} = V_{e,tot} + V_{he,tot} - V_{e,l} \quad V_{e,l} = L_e A_e \beta_e \quad (112)$$

The vapour volume  $V_{e,v}$  and the liquid volume  $V_{e,l}$  are given by eq. (112), with  $V_{e,tot}$  for the total volume of the evaporator and  $V_{he,tot}$  for the total volume of the internal heat exchanger.

The evaporation mass flow  $m_e^*$  is directly proportional to the difference of the evaporation pressure  $p_{e,l}$  in the liquid phase and the pressure  $p_{e,v}$  in the vapour phase (eq. (113)). The mass transfer rate  $k_{tr,e}$  is regarded as a parameter.

$$m_e^* = k_{tr,e} \cdot (p_{e,l} - p_{e,v}) \quad (113)$$

### 11.1.2 Superheating

The superheating process takes place in the region after the evaporation length  $L_e$ , where there is no liquid left. The energy balance for the superheating process is given by (eq. (114)) with the superheating temperature  $T_{ov}$ . Therefore, the convective heat flow  $Q_{ov}^*$  as well as the transfer heat flow  $Q_{e,ov}^*$  in the evaporator and the transfer heat flow  $Q_{he,ov}^*$  in the internal heat exchanger have to be calculated.

$$\frac{dT_{ov}}{dt} \cdot m_{ov} c_{ov} = \left[ Q_{ov}^* + Q_{he,ov}^* + Q_{e,ov}^* \right] \quad (114)$$

The convective heat flow  $Q_{ov}^*$  is the sum of several heat flows, depending on the operation mode of the heat pump. Eq. (115) is valid for the normal operating mode, with the saturated vapour enthalpy  $h''(p_{e,l})$  of the evaporated refrigerant at the input and the enthalpy  $h_v(T_{ov}, p_{e,v})$  of the superheated refrigerant at the output.

$$Q_{ov,1}^* = u_{cmp} \cdot m_e^* \cdot h''(p_{e,l}) - u_{cmp} \cdot m_{cmp}^* \cdot h_v(T_{ov}, p_{e,v}) \quad (115)$$

For the defrosting mode, eq. (116) considers the enthalpy of a part  $m_{hg,g}^*$  of the hot gas mass flow, which is directly feeding the gaseous phase.

$$Q_{ov,2}^* = m_{hg,g}^* \cdot h_v(T_{cmp, hp}, p_{c,v}) \quad (116)$$

For the shutdown sequence, when the compressor is switched off ( $u_{cmp} = 0$ ), the first term of eq. (117) considers the backward flow through the compressor ( $m_{cmp}^* < 0$ ). Then, the enthalpy  $h_v(T_{cmp}, p_{e,v})$  at the low pressure side of the compressor is an input. The second term describes the case of a negative mass flow  $m_e^* < 0$ , which is condensating.

$$Q_{ov,0}^* = -(1 - u_{cmp}) \cdot m_{cmp}^* \cdot h_v(T_{cmp}, p_{e,v}) + (1 - u_{cmp}) \cdot m_e^* \cdot h_v(T_{ov}, p_{e,v}) \quad (117)$$

These heat flows are summarized in eq. (118).

$$Q_{ov}^* = Q_{ov,0}^* + Q_{ov,1}^* + Q_{ov,2}^* \quad (118)$$

The transfer heat flow  $Q_{he,ov}^*$  in eq. (114) between the subcooling process and the superheating process is given by eq. (119), with the mean subcooling temperature  $\bar{T}_{sc}$  and the mean superheating temperature  $\bar{T}_{ov}$ . The heat transfer parameter  $kb_{he,ov}$  is defined for the region  $L_e..L_{he,tot}$  in the internal heat exchanger (if  $L_e \geq L_{e,tot}$ ).

$$Q_{he,ov}^* = kb_{he,ov} \cdot L_{he,ov} \cdot (\bar{T}_{sc} - \bar{T}_{ov}) \quad (119)$$

Eq. (120) describes the heat flow  $Q_{e,ov}^*$  in eq. (114) transferred from the air to the superheating process. The heat transfer parameter  $k_{be,ov}$  is defined for the region  $L_e..L_{e,tot}$ , if the liquid is completely evaporated in the evaporator ( $L_e < L_{e,tot}$ ).

$$Q_{e,ov}^* = kb_{e,ov} \cdot L_{e,ov} \cdot (\bar{T}_a - \bar{T}_{ov}) \quad (120)$$

### 11.1.3 Compression

The mass flow for the normal operating mode is calculated by the characteristics of the compressor (eq. (121)) as a function of the pressure ratio  $p_{c,v}/p_{e,v}$ . On the low-pressure side, the pressure is equal to the evaporation pressure  $p_{e,v}$ , and on the high-pressure side, the pressure is equal to the condensation pressure  $p_{c,v}$ . The parameters  $a_{cmp}$ ,  $b_{cmp}$  and  $d_{cmp}$  describe the pressure-to-mass-flow characteristics, and the parameter  $\lambda_{cmp}$  is defined as the mass flow supply efficiency of the compressor.

$$m_{cmp,1}^* = u_{cmp} \cdot \lambda_{cmp} \cdot \left[ a_{cmp} \cdot \left( \frac{p_{c,v}}{p_{e,v}} \right)^{b_{cmp}} - d_{cmp} \right] \quad (121)$$

For the shut-down sequence with  $u_{cmp} = 0$ , the compressor is regarded as a throttle for which eq. (122) is used with the valve constant  $k_{cmp}$ . The migration process of the refrigerant is mainly caused by a backflow in the compressor, presuming that the liquid valve before the expansion valve is closed during the shut-down sequence ( $u_{lv} = 0$ ).

$$m_{cmp,0}^* = -(1 - u_{cmp}) \cdot k_{cmp} \cdot \sqrt{p_{c,v} - p_{e,v}} \quad (122)$$

These mass flows are summarized in eq. (123).

$$m_{cmp}^* = m_{cmp,0}^* + m_{cmp,1}^* \quad (123)$$

The temperature  $T_{cmp,hp}$  on the high-pressure side is calculated from the temperature  $T_{ov}$  at the suction line by the polytropic law (eq. (124)), with the polytropic exponent  $c_{cmp}$ .

$$T_{cmp,hp} = T_{ov} \cdot \left( \frac{p_{c,v}}{p_{e,v}} \right)^{c_{cmp}} \quad (124)$$

From the enthalpy difference between the input and the output of the compressor, the electrical power input is calculated by eq. (125) with the electro-mechanical efficiency  $\eta_{cmp}$ .

$$P_{HP} = \frac{m_{cmp}^*}{\eta_{cmp}} \cdot [h_v(T_{cmp, hp}, p_{c,v}) - h_v(T_{ov}, p_{e,v})] \quad (125)$$

Eq. (126) considers the dynamics of the compressor as they influence the compressor temperature  $T_{cmp}$ .

$$\frac{dT_{cmp}}{dt} \cdot m_{cmp} c_{cmp} = Q_{cmp}^* \quad (126)$$

Again, the convective heat flow  $Q_{cmp}^*$  is the sum of several heat flows. For the normal operating mode ( $u_{cmp} = 1$ ,  $u_{hgv} = 0$ ), eq. (127) describes the difference between the enthalpy  $h_v(T_{cmp, hp}, p_{c,v})$ , which is given by the statically calculated output temperature  $T_{cmp, hp}$  of eq. (124), and the enthalpy  $h_v(T_{cmp}, p_{c,v})$ , which is defined by the state variable  $T_{cmp}$ .

$$Q_{cmp,1}^* = (u_{cmp} - u_{hgv}) \cdot m_{cmp}^* \cdot [h_v(T_{cmp, hp}, p_{c,v}) - h_v(T_{cmp}, p_{c,v})] \quad (127)$$

For the shut-down sequence with  $u_{cmp} = 0$ , a backward mass flow  $m_{cmp}^* < 0$  is taken into account (eq. (128)). Here the enthalpy  $h_v(T_{c,g}, p_{c,v})$ , coming from the gaseous phase of the condenser, is an input.

$$Q_{cmp,0}^* = -(1 - u_{cmp}) \cdot m_{cmp}^* \cdot [h_v(T_{c,g}, p_{c,v}) - h_v(T_{cmp}, p_{c,v})] \quad (128)$$

The sum of the heat flows is expressed by eq. (129).

$$Q_{cmp}^* = Q_{cmp,0}^* + Q_{cmp,1}^* \quad (129)$$

### 11.1.4 Hot Gas Cooling

The hot gas cooling process takes place in the upper region of the condenser (above  $L_c$ ). The energy balance is given by eq. (130) with the gas temperature  $T_{c,g}$ , the convective heat flow  $Q_{hgc}^*$  and the transfer heat flow  $Q_{c,g}^*$ .

$$\frac{dT_{c,g}}{dt} \cdot m_{c,g} c_{c,g} = [Q_{hgc}^* - Q_{c,g}^*] \quad (130)$$

The convective heat flow  $Q_{hgc}^*$  is the sum of several heat flows. For normal operation in eq. (131), the enthalpy  $h_v(T_{cmp}, p_{c,v})$  from the compressor is an input and the enthalpy  $h_v(T_{c,g}, p_{c,v})$  of the cooled gas is an output. During the defrosting cycle, the hot gas mass flow  $m_{hg}^*$  is bypassing the condenser and

therefore is subtracted from the compressor mass flow  $m_{cmp}^*$ . The mass flow  $m_c^*$  will condensate later.

$$Q_{hgc,1}^* = u_{cmp} \cdot (m_{cmp}^* - m_{hg}^*) \cdot h_v(T_{cmp}, p_{c,v}) - u_{cmp} \cdot m_c^* \cdot h_v(T_{c,g}, p_{c,v}) \quad (131)$$

For the shut-down sequence with  $u_{cmp} = 0$ , eq. (132) considers the backward mass flows  $m_c^* < 0$  and  $m_{cmp}^* < 0$ . In this case, the mass flow  $m_c^* < 0$  is evaporating from the liquid phase.

$$Q_{hgc,0}^* = -(1 - u_{cmp}) \cdot m_c^* \cdot h''(p_{c,l}) + (1 - u_{cmp}) \cdot m_{cmp}^* \cdot h_v(T_{c,g}, p_{c,v}) \quad (132)$$

The heat flows are summarized in eq. (133).

$$Q_{hgc}^* = Q_{hgc,0}^* + Q_{hgc,1}^* \quad (133)$$

The transfer heat flow  $Q_{c,g}^*$  (cf. eq. (130)) between the hot gas side and the water side is given by eq. (134), with the mean gas temperature  $\bar{T}_{c,g}$  and the mean water temperature  $\bar{T}_w$ . The heat transfer parameter  $kb_{c,g}$  is defined for the gas region  $L_c \cdot L_{c,tot}$ .

$$Q_{c,g}^* = kb_{c,g} \cdot L_{c,g} \cdot (\bar{T}_{c,g} - \bar{T}_w) \quad (134)$$

### 11.1.5 Condensation

The gaseous refrigerant condensates between the plates of the condenser. Because of gravity, the liquid phase fills up the condenser from the bottom to the level  $L_c$ . With a changing filling grade of the condenser,  $L_c$  is time-varying. The balance for the liquid mass  $m_{c,l}$  is given by eq. (135), with  $m_c^*$  representing the condensating input mass flow and  $m_{exp}^*$  representing the output mass flow to the expansion valve (cf. eq. (155)).

$$\frac{dm_{c,l}}{dt} = m_c^* - m_{exp}^* \quad (135)$$

Using the derivative  $dm_{c,l}/dt$  of eq. (135), the differential equation for  $L_c$  is built (eq. (136)) with the liquid density  $\rho'(p_{c,l})$  and the condenser cross-sectional area  $A_c$ . Regarding a condenser with plates,  $A_c$  is the summarized cross-sectional area between all parallel plates, defined by the distance between the plates and the width of the plates.

$$\frac{dL_c}{dt} = \frac{dm_{c,l}}{dt} \cdot \frac{1}{\rho'(p_{c,l}) \cdot A_c} \quad (136)$$

The liquid level  $L_c$  defines the height of the liquid phase  $L_{c,c}$  and the height of the gaseous phase  $L_{c,g}$  (eq. (137)). When  $L_c$  reaches the limit  $L_{c,limit}$ , the condenser is full.

$$\begin{pmatrix} L_{c,c} \\ L_{c,g} \end{pmatrix} = \begin{pmatrix} L_c \\ L_{c,tot} - L_c \end{pmatrix} \quad (0 \leq L_c \leq L_{c,limit}) \quad (137)$$

Eq. (138) describes the balance for the vapour mass  $m_{c,v}$  with the input mass flow ( $m_{cmp}^* - m_{hg}^*$ ), the condensating mass flow  $m_c^*$  and the gaseous output mass flow  $m_{exp,g}^*$ . During the defrosting cycle, the bypassed hot gas mass flow  $m_{hg}^*$  is subtracted from the input mass flow  $m_{cmp}^*$  coming from the compressor. The gaseous output mass flow  $m_{exp,g}$  (eq. (154)) is nonzero only for the case of an almost “empty” condenser ( $L_c \rightarrow 0$ ).

$$\frac{dm_{c,v}}{dt} = (m_{cmp}^* - m_{hg}^*) - m_c^* - m_{exp,g}^* \quad (138)$$

The energy balance for the liquid phase is expressed by eq. (139), with the time derivative of the pressure  $dp_{c,l} / dt$  and the partial derivative of the energy  $du'_c / dp_{c,l}$ , subject to the condensation pressure  $p_{c,l}$  of the liquid phase.

$$\frac{dp_{c,l}}{dt} \frac{du'_c}{dp_{c,l}} m_{c,l} + u'_c \frac{dm_{c,l}}{dt} = [Q_{c,l}^* - Q_{c,c}^*] \quad (139)$$

During normal operation with  $u_{cmp} = 1$ , eq. (140) represents the convective heat flow  $Q_{c,l}^*$ . The condensating mass flow  $m_c^*$  with the vapour enthalpy  $h_v(T_{c,g}, p_{c,v})$  is an input, whereas the mass flow  $m_{exp}^*$  with the saturated liquid enthalpy  $h'(p_{c,l})$  is an output.

$$Q_{c,l,1}^* = u_{cmp} \cdot m_c^* \cdot h_v(T_{c,g}, p_{c,v}) - m_{exp}^* \cdot h'(p_{c,l}) \quad (140)$$

For the shut-down sequence with  $u_{cmp} = 0$ , eq. (141) considers a negative mass flow  $m_c^* < 0$ , which is evaporating. Then the saturated vapour enthalpy  $h''(p_{c,l})$  is an output.

$$Q_{c,l,0}^* = (1 - u_{cmp}) \cdot m_c^* \cdot h''(p_{c,l}) \quad (141)$$

The total heat flow  $Q_{c,l}^*$  is defined by eq. (142).

$$Q_{c,l}^* = Q_{c,l,0}^* + Q_{c,l,1}^* \quad (142)$$

The transfer heat flow  $Q_{c,c}^*$  in eq. (139) between the liquid phase of the condenser and the water side is driven by the difference between the mean liquid

refrigerant temperature  $\bar{T}_{c,l}$  and the mean water temperature  $\bar{T}_w$  (eq. (143)), with the heat transfer parameter  $kb_{c,c}$  for the region  $0..L_c$ .

$$Q_{c,c}^* = kb_{c,c} \cdot L_{c,c} \cdot (\bar{T}_{c,l} - \bar{T}_w) \quad (143)$$

In the vapour phase, the pressure  $p_{c,v}$  is calculated from the gas equation (144), with the temperature  $T_{c,g}$  and the specific volume  $v_{c,v}$  as inputs.

$$p_{c,v} = p_{c,v}(T_{c,g}, v_{c,v}) \quad v_{c,v} = \frac{V_{c,v}}{m_{c,v}} \quad (144)$$

$$V_{c,v} = V_{c,tot} - V_{c,l} \quad V_{c,l} = L_c A_c \quad (145)$$

The vapour volume  $V_{c,v}$  and the liquid volume  $V_{c,l}$  are given by eq. (145), with  $V_{c,tot}$  for the total volume of the refrigerant side in the condenser.

The condensation mass flow  $m_c^*$  is directly proportional to the difference between the pressure  $p_{c,v}$  in the vapour phase and the condensation pressure  $p_{c,l}$  in the liquid phase (eq. (146)), with the mass transfer rate  $k_{tr,c}$ .

$$m_c^* = k_{tr,c} \cdot (p_{c,v} - p_{c,l}) \quad (146)$$

### 11.1.6 Subcooling

The subcooling process takes place in the internal heat exchanger. The energy balance is given by eq. (147), with the subcooling temperature  $T_{sc}$ , the convective heat flow  $Q_{sc}^*$ , and the transfer heat flows  $Q_{he,e}^*$ ,  $Q_{he,ov}^*$  to the evaporation and the superheating sides (cf. eqs. (109), (119)).

$$\frac{dT_{sc}}{dt} \cdot m_{sc} c_{sc} = [Q_{sc}^* - Q_{he,e}^* - Q_{he,ov}^*] \quad (147)$$

Eq. (148) is valid for the normal operating mode, with the saturated liquid enthalpy  $h'(p_{c,l})$  from the condenser at the input and the enthalpy  $h_l(T_{sc})$  of the subcooled refrigerant at the output.

$$Q_{sc,1}^* = m_{exp}^* \cdot [h'(p_{c,l}) - h_l(T_{sc})] \quad (148)$$

For the shut-down sequence, eq. (149) reflects the heat transfers enforced by the difference between the condenser bubble-point temperature  $T_b(p_{c,l})$  and the subcooling temperature  $T_{sc}$ . Here the mass flow is (almost) zero, since the liquid valve is closed ( $u_{lv} = 0$ ). Therefore the convective parameter  $cm_{sc,min}^*$  is introduced.

$$Q_{sc,0}^* = cm_{sc,\min}^* \cdot (T_b(p_{c,l}) - T_{sc}) \quad (149)$$

These heat flows are summarized in eq. (150).

$$Q_{sc}^* = Q_{sc,0}^* + Q_{sc,1}^* \quad (150)$$

### 11.1.7 Expansion

A thermostatic expansion valve works like a P-controller, which feeds back the superheating  $\Delta T_{ov}$  and adjusts the mass flow  $m_{exp}^*$  to the evaporator (cf. Section 4.1). In eq. (151), the variable  $T_d(p_{e,l})$  is the dew-point temperature of the evaporation process.

$$\Delta T_{ov} = T_{ov} - T_d(p_{e,l}) \quad (151)$$

In reality, the valve opening is set by the difference between the pressure  $p_{ov}(T_{ov})$  and the pressure  $p_{e,l}$ . In the model, the temperature difference  $\Delta T_{ov}$  is used directly to calculate the valve opening (152). The range for the valve opening  $u_{exp}$  is between 0 for a closed valve and 1 for an open valve.

$$u_{exp} = a_{exp} \cdot (\Delta T_{ov} - \Delta T_{ov,0}) + b_{exp} \quad 0 \leq u_{exp} \leq 1 \quad (152)$$

With a superheating setpoint  $\Delta T_{ov,0}$ , the parameter  $a_{exp}$  corresponds to the P value of the controller, whereas the parameter  $b_{exp}$  defines the opening offset for the operating point. From the valve opening  $u_{exp}$  and the pressure drop over the valve, the mass flow  $m_{exp}^*$  through the valve is calculated (eqs. (153) ..(155)). Here the characteristics correspond to an incompressible flow. Eq. (153) calculates the liquid mass flow  $m_{exp,l}^*$  from the liquid condensation pressure  $p_{c,l}$  and the liquid evaporation pressure  $p_{e,l}$ . The gaseous mass flow  $m_{exp,g}^*$  is calculated by eq. (154) from the vapour condensation pressure  $p_{c,v}$  and the vapour evaporation pressure  $p_{e,v}$ . The total mass flow is given by eq. (155).

$$m_{exp,l}^* = k_l \cdot u_{lv} \cdot u_{exp} \cdot k_{exp,l} \sqrt{p_{c,l} - p_{e,l}} \quad (153)$$

$$m_{exp,g}^* = (1 - k_l) \cdot u_{lv} \cdot u_{exp} \cdot k_{exp,g} \sqrt{p_{c,v} - p_{e,v}} \quad (154)$$

$$m_{exp}^* = m_{exp,l}^* + m_{exp,g}^* \quad (155)$$

The liquid ratio  $k_l$  depends on the liquid level  $L_c$  in the condenser (eq. (156)). Only for low levels  $L_c \leq L_{c,\min}$ , the gaseous mass flow  $m_{exp,g}^*$  is taken into account ( $k_l < 1$ ).

$$k_l = \begin{cases} \frac{L_c}{L_{c,\min}} & (L_c \leq L_{c,\min}) \\ 1 & (L_c > L_{c,\min}) \end{cases} \quad (156)$$

For the the shut-down sequence or the defrosting sequence, the mass flows  $m_{exp,l}^*$  and  $m_{exp,g}^*$  are set to zero, since the magnetic liquid valve is closed ( $u_{lv} = 0$ ). For the defrosting mode, eq. (157) defines the mass flow  $m_{hg}^*$  through the hot-gas valve. The mass flow  $m_{hg}^*$  is separated into a condensating part  $m_{hg,c}^*$  and a gaseous part  $m_{hg,g}^*$  (eq. (158)).

$$m_{hg}^* = u_{hgv} \cdot k_{hg} \sqrt{p_{c,v} - p_{e,v}} \quad (157)$$

$$m_{hg,c}^* = k_{hg,c} m_{hg}^* \quad m_{hg,g}^* = (1 - k_{hg,c}) \cdot m_{hg}^* \quad (158)$$

During the expansion, the entropy  $s$  is increasing, but the enthalpy  $h$  is constant. Thus the enthalpy  $h_{exp,l}$  at the output of the expansion valve is equal to the enthalpy  $h_l(T_{sc})$  at the input, which is a function of the subcooler output temperature  $T_{sc}$ .

$$h_{exp,l} = h_l(T_{sc}) \quad (159)$$

### 11.1.8 Air Side

The energy balance for the heat source is defined by eq. (160), with the output air temperature  $T_{a,o}$ . The transfer heat flows  $Q_{e,e}^*$  and  $Q_{e,ov}^*$  to the evaporator are taken from eq. (108) and eq. (120).

$$\frac{dT_{a,o}}{dt} \cdot m_a c_a = [Q_a^* - Q_{e,e}^* - Q_{e,ov}^*] \quad (160)$$

The convective heat flow  $Q_{a,o}^*$  depends on the air mass flow  $m_a^*$  and on the difference between the input and output temperatures  $T_{a,i}$ ,  $T_{a,o}$  (eq. (161)). For the shut-down sequence, when the air fan is not running ( $m_a^* = 0$ ),  $m_{a,\min}^*$  represents a minimal convection.

$$Q_{a,o}^* = (m_a^* + m_{a,\min}^*) \cdot c_{p,a} \cdot (T_{a,i} - T_{a,o}) \quad (161)$$

When the air fan is running ( $u_{fan} = 1$ ), the air mass flow is assumed to be a constant parameter  $m_{a,0}^*$  (eq. (162)).

$$m_a^* = u_{fan} \cdot m_{a,0}^* \quad (162)$$

### 11.1.9 Water Side

Eq. (163) defines the energy balance for the heat sink, with the output water temperature  $T_{w,o}$ . The transfer heat flows  $Q_{c,c}^*$  and  $Q_{c,g}^*$  to the condenser are defined by eq. (143) and eq. (134).

$$\frac{dT_{w,o}}{dt} \cdot m_w c_w = \left[ Q_w^* + Q_{c,c}^* + Q_{c,g}^* \right] \quad (163)$$

The convective heat flow  $Q_w^*$  depends on the water mass flow  $m_w^*$ , the input temperature  $T_{w,i}$ , and the output temperature  $T_{w,o}$  (eq. (164)), with a minimal convection  $m_{w,min}^*$  for the shut-down sequence.

$$Q_w^* = (m_w^* + m_{w,min}^*) \cdot c_{p,w} \cdot (T_{w,i} - T_{w,o}) \quad (164)$$

When the circulating pump is running ( $u_{pump} = 1$ ), the water flow is assumed to be a constant parameter  $m_{w,0}^*$  (eq. (162)).

$$m_w^* = u_{pump} \cdot m_{w,0}^* \quad (165)$$

In the definition of the heat output  $Q_{HP}^*$  (eq. (166)),  $m_{w,min}^*$  is set to zero, because the corresponding heat flow is lost to the „environment“ when the circulating pump is not running.

$$Q_{HP}^* = m_w^* \cdot c_{p,w} \cdot (T_{w,i} - T_{w,o}) \quad (166)$$

With the heat output  $Q_{HP}^*$  of eq. (166) and the electric power input  $P_{HP}$  of eq. (125), the coefficient of performance (COP) is defined by eq. (167).

$$COP(t) = \frac{Q_{HP}^*(t)}{P_{HP}(t)} \quad (167)$$

### 11.1.10 Icing of the Evaporator

At low temperatures and high humidity of the outside air, some icing effect takes place on the evaporator. This effect is modelled in [Ginsburg 99], which has been slightly simplified and modified for the present model. From the relative humidity  $\varphi$  of the air, the vapour mass ratio  $x$  can be calculated (eq. (168)). The vapour mass  $m_v$  as well as the whole air mass  $m_a$  are substituted by the corresponding gas equations. The ratio of the air gas constant  $R_a$  and the vapour gas constant  $R_v$  is defined by the parameter  $k_R$ . The partial pressure of the air  $p_a$  is expressed by  $(p_o - p_v)$ , with the overall pressure  $p_o$  and the partial pressure  $p_v$  of the vapour. The relative humidity  $\varphi$  is given as the ratio of  $p_v$  and the saturation pressure  $p_s$ .

$$x = \frac{m_v}{m_a} = \frac{\frac{p_v \cdot V_0}{R_v \cdot T}}{\frac{p_a \cdot V_0}{R_a \cdot T}} = \frac{R_a}{R_v} \cdot \frac{p_v}{p_a} = k_R \cdot \frac{p_v}{p_0 - p_v} \quad \varphi = \frac{p_v}{p_s} \quad (168)$$

Substituting  $p_v$  by  $\varphi \cdot p_s$  yields eq. (169).

$$x(\varphi) = \frac{k_R}{\frac{p_0}{\varphi \cdot p_s} - 1} \quad (169)$$

Eq. (169) is evaluated for the evaporator air input, with the humidity  $\varphi_{a,i}$  and the temperature  $T_{a,i}$  (eq. (170)).

$$x_{a,i} = x(\varphi_{a,i}) = \frac{k_R}{\frac{p_0}{\varphi_{a,i} \cdot p_s(T_{a,i})} - 1} \quad (170)$$

The saturated vapour mass ratio  $x_{sa,o}$  at the output is given by eq. (171) for  $\varphi=1$ . The oversaturated part  $x_{a,i}-x_{sa,o}$  feeds the icing process.

$$x_{sa,o} = x(\varphi = 1) = \frac{k_R}{\frac{p_0}{p_s(T_{a,o})} - 1} \quad (171)$$

Using these equations, the preconditions for icing and the rate of icing can be determined (case I, eqs. (172) and (173)). The mass flow  $m_{ice,build}^*$ , which is proportional to the air mass flow  $m_a^*$ , increases the ice layer, whereas the heat flow  $Q_{ice,build}^*$  arises from the enthalpy change  $r_{ice}$  of the icing process. Icing is only possible for temperatures below  $T_{ice,build}$  ( $1 \text{ }^\circ\text{C}$  is chosen for the simulations). The factor  $k_{ice,build}$  is regarded as a parameter.

*case I: ( $x_{a,i} > x_{sa,o}$ ) and ( $T_{a,o} < T_{ice,build}$ )  $\Rightarrow$  ice building*

$$m_{ice,build}^* = k_{ice,build} \cdot m_a^* \cdot (x_{a,i} - x_{sa,o}) \quad (172)$$

$$Q_{ice,build}^* = m_{ice,build}^* \cdot r_{ice} \quad (173)$$

For temperatures above  $T_{ice,selfdefr}$  (here  $0 \text{ }^\circ\text{C}$  is chosen), some self-defrosting process is initiated (case II, eqs. (174) and (175)), which decreases the ice layer by the mass flow  $m_{ice,selfdefr}^*$ . The factor  $k_{ice,selfdefr}$  is regarded as a parameter.

case II:  $(T_{a,o} > T_{ice,selfdefr}) \Rightarrow$  self-defrosting

$$m_{ice,selfdefr}^* = k_{ice,mass} \cdot k_{ice,selfdefr} \cdot (T_{a,o} - T_{ice,selfdefr})^2 \quad (174)$$

$$Q_{ice,selfdefr}^* = m_{ice,selfdefr}^* \cdot r_{ice} \quad (175)$$

The factor  $k_{ice,mass}$  considers the actual ice mass  $m_{ice}$  and decreases the self-defrosting process for small values  $m_{ice} < m_{ice,min}$  (eq. (176)).

$$k_{ice,mass} = \begin{cases} \frac{m_{ice}}{m_{ice,min}} & m_{ice} < m_{ice,min} \\ 1 & m_{ice} \geq m_{ice,min} \end{cases} \quad (176)$$

During the defrosting sequence the hot-gas valve is opened, the evaporator warms up, and the ice melts (case III, eq. (177) and (178)). It is assumed that the temperature of the refrigerant side of the evaporator rises above the outside air temperature and that a part of the heat flow  $(Q_{e,e}^* + Q_{e,ov}^*) < 0$  to the air side is directly used to melt the ice (eq. (177)). The enforced defrosting mass flow  $m_{ice,enfdefr}^*$  is thus calculated from the heat flow  $Q_{ice,enfdefr}^*$  and the enthalpy change  $r_{ice}$  for liquefaction (eq. (178)).

case III: defrosting cycle activated  $\Rightarrow$  enforced defrosting

$$Q_{ice,enfdefr}^* = -k_{ice,mass} \cdot k_{ice,enfdefr} \cdot (Q_{e,e}^* + Q_{e,ov}^*) \quad (177)$$

$$m_{ice,enfdefr}^* = \frac{Q_{ice,enfdefr}^*}{r_{ice}} \quad (178)$$

With the ice mass flows, the ice mass balance is defined by eq. (179).

$$\frac{dm_{ice}}{dt} = m_{ice,build}^* - m_{ice,selfdefr}^* - m_{ice,enfdefr}^* \quad (179)$$

An existing ice layer causes several effects, the first of which is a pressure drop  $\Delta p_a$  on the air side of the evaporator (eq. (180)), with the parameters  $\Delta p_{a,0}$  and  $k_{\Delta p}$ .

$$\Delta p_a = \Delta p_{a,0} + k_{\Delta p} \cdot m_{ice} \quad (180)$$

Secondly, the heat transfer rate of the evaporator is reduced. The reduction factor  $k_{ice}$  is introduced by eq. (181), with the parameters  $a_{ice}$  and  $m_{ice,min}$ .

$$k_{ice} = 1 - a_{ice} \cdot (m_{ice} - m_{ice,min}) \quad (181)$$

The evaporation transfer heat flow  $Q_{e,e}^*$  of eq. (108) is replaced by eq. (182), and the superheating transfer heat flow  $Q_{e,ov}^*$  of eq. (120) is replaced by eq. (183).

$$Q_{e,e}^* = kb_{e,e} \cdot k_{ice} \cdot L_{e,e} \cdot (\bar{T}_a - \bar{T}_{e,l}) \quad (182)$$

$$Q_{e,ov}^* = kb_{e,ov} \cdot k_{ice} \cdot L_{e,ov} \cdot (\bar{T}_a - \bar{T}_{ov}) \quad (183)$$

Finally, the additional icing heat flows are added to eq. (160), which results in eq. (184).

$$\frac{dT_{a,o}}{dt} \cdot m_a c_a = \left[ Q_a^* - Q_{e,e}^* - Q_{e,ov}^* + Q_{ice,build}^* - Q_{ice,selfdefr}^* - Q_{ice,enfdefr}^* \right] \quad (184)$$

### 11.1.11 Refrigerant Data

The physical property data of the refrigerant R407C is defined by polynomials with validated parameters from the refrigerant manufacturer [Klea 95]. The values of the parameters used below are listed in Appendix C, Section ‘Simulation Model Parameters’, ‘Refrigerant Data R407C’. For the equation of state, the Martin-Hou approach is used (eq. (185)), with the relative states  $(T_r, p_r, v_r)$  regarding the critical point  $(T_c, p_c, v_c)$  (eq. (186)).

$$p_r(T_r, v_r) = \frac{X T_r}{v_r - B} + \sum_{i=1}^4 \frac{A_i + B_i T_r + C_i e^{(-K T_r)}}{(v_r - B)^{i+1}} \quad (185)$$

$$T_r = T/T_c \quad p_r = p/p_c \quad v_r = v/v_c = v\rho_c \quad (186)$$

By only considering the first term of the sum ( $i=1$ ) and with  $B=0$ , the equation can be simplified to eq. (187) without a significant lack of precision for the region required. This equation is used in eqs. (111) and (144).

$$p(T, v) = \left( \frac{X T/T_c}{v/v_c} + \frac{A_1 + B_1 T/T_c + C_1 e^{(-KT/T_c)}}{(v/v_c)^2} \right) \cdot p_c \quad (187)$$

Because of the azeotropic properties of R407C, there is a temperature glide during the evaporation. For a given evaporation pressure  $p$ , a bubble-point temperature  $T_b$ , a mid-point temperature  $T_m$ , and a dew-point temperature  $T_d$  are defined (eqs. (188), (189) and (190)).

$$T_b(p) = A_b + B_b \ln p + C_b (\ln p)^2 + D_b (\ln p)^3 \quad (188)$$

$$T_m(p) = A_m + B_m \ln p + C_m (\ln p)^2 + D_m (\ln p)^3 \quad (189)$$

$$T_d(p) = A_d + B_d \ln p + C_d (\ln p)^2 + D_d (\ln p)^3 \quad (190)$$

The saturated liquid enthalpy  $h'(p)$  is defined by eq. (191), with eq. (192) as a function of the pressure  $p$ .

$$h'(p) = A' + B'x_b(p) + C'x_b^2(p) + D'x_b^3(p) + E'x_b^4(p) \quad (191)$$

$$x_b(p) = (1 - T_b(p)/T_c)^{1/3} \quad (192)$$

In a similar way the latent heat of vaporization  $r(p)$  is defined by eq. (193) with eq. (194).

$$r(p) = A_r + B_rx_m(p) + C_rx_m^2(p) + D_rx_m^3(p) + E_rx_m^4(p) \quad (193)$$

$$x_m(p) = (1 - T_m(p)/T_c)^{1/3} \quad (194)$$

The saturated vapour enthalpy  $h''(p)$  (eq. (195)) is the sum of eq. (191) and eq. (193).

$$h''(p) = h'(p) + r(p) \quad (195)$$

Eq. (196) defines the liquid enthalpy  $h_l(T_b)$  and is similar to eq. (191), but here the enthalpy is defined as a function of the temperature  $T_b$ .

$$h_l(T_b) = A' + B'x_b(T_b) + C'x_b^2(T_b) + D'x_b^3(T_b) + E'x_b^4(T_b) \quad (196)$$

$$x_b(T_b) = (1 - T_b/T_c)^{1/3} \quad (197)$$

The gas heat capacity  $c_p$  is a function of the gas temperature  $T$  (eq. (198)).

$$c_p(T) = A_c + B_cT + C_cT^2 + D_c/T \quad (198)$$

The heat capacity  $c_p(T)$  is also used to calculate the vapour or gas enthalpy  $h_v(T,p)$  from  $h''(p)$  and the temperature offset  $(T - T_d(p))$  (eq. (199)). Therefore, the function  $c_p(T)$  is evaluated at the mid-temperature between  $T_d(p)$  and  $T$ .

$$h_v(T, p) = h''(p) + c_p\left(\frac{T_d(p) + T}{2}\right) \cdot (T - T_d(p)) \quad (199)$$

Eqs. (200) and (201) represent the polynomials for the liquid density  $\rho'(p)$  and the saturated vapour density  $\rho''(p)$ .

$$\rho'(p) = A'_\rho + B'_\rho x_b(p) + C'_\rho x_b^2(p) + D'_\rho x_b^3(p) + E'_\rho x_b^4(p) \quad (200)$$

$$\rho''(p) = A''_{\rho} + B''_{\rho} x_d(p) + C''_{\rho} x_d^2(p) + D''_{\rho} x_d^3(p) + E''_{\rho} x_d^4(p) \quad (201)$$

### 11.1.12 Mean Temperatures and Simplifications

For the the heat transfer equations, the mean temperatures  $\bar{T}_i$  have to be calculated. In the evaporation heat transfer equation of eq. (108), the mean air temperature  $\bar{T}_a$  is a function of the air inlet and outlet temperatures  $T_{a,i}$  and  $T_{a,o}$  (eq. (202), left side).

$$\bar{T}_a = f_a(T_{a,i}, T_{a,o}) \quad \bar{T}_a = T_{a,o} \quad (202)$$

Often, logarithmic functions are chosen. In order to reduce computational time and to prevent numerical problems, here linear functions are implemented. The mean air temperature  $\bar{T}_a$  is simply defined by the air outlet temperature  $T_{a,o}$  (eq. (202), right side). The mean temperature  $\bar{T}_{e,l}$  in eq. (108) is either defined by the evaporation mid-point temperature  $T_m(p_{e,l})$  for normal operation with  $u_{cmp} = 1$  or by the dew-point temperature  $T_d(p_{e,l})$  for the shut-down sequence with  $u_{cmp} = 0$  (eq. (203)).

$$\bar{T}_{e,l} = u_{cmp} \cdot T_m(p_{e,l}) + (1 - u_{cmp}) \cdot T_d(p_{e,l}) \quad (203)$$

In a similar way, the mean temperatures in eqs. (109), (119), (120), (134), (143) are defined by eqs. (204) to (206).

$$\bar{T}_{sc} = T_{sc} \quad \bar{T}_{ov} = T_{ov} \quad (204)$$

$$\bar{T}_{c,g} = T_{c,g} \quad \bar{T}_w = T_{w,o} \quad (205)$$

$$\bar{T}_{c,l} = u_{cmp} \cdot T_m(p_{c,l}) + (1 - u_{cmp}) \cdot T_b(p_{c,l}) \quad (206)$$

As another simplification, the thermal inertiae in the energy balance equations (114), (126), (130), (147), (160), (163) are regarded as constant parameters  $\tau_i$ , which are defined in the eqs. (207) and (208).

$$\tau_{ov} = m_{ov} \cdot c_{ov} \quad \tau_{cmp} = m_{cmp} \cdot c_{cmp} \quad \tau_{c,g} = m_{c,g} \cdot c_{c,g} \quad (207)$$

$$\tau_{sc} = m_{sc} \cdot c_{sc} \quad \tau_a = m_a \cdot c_a \quad \tau_w = m_w \cdot c_w \quad (208)$$

In analogy, the thermal inertia in the evaporator energy balance (eq. (103)) is regarded as a constant parameter  $\tau_{e,l}$ . Therefore, the assumptions of eq. (209) have to be made, which are not straightforward. Some numerical simulations for the case with and without these simplifications have shown that only a small error is produced by using eq. (209).

$$\tau_{e,l} = \frac{du'_e}{dp_{e,l}} m_{e,l} \approx \text{const.} \quad u'_e \frac{dm_{e,l}}{dt} \approx 0 \quad (209)$$

For the thermal inertia in the condenser energy balance (eq. (139)), similar investigations lead to eq. (210).

$$\tau_{c,l} = \frac{du'_c}{dp_{c,l}} m_{c,l} \approx \text{const.} \quad u'_c \frac{dm_{c,l}}{dt} \approx 0 \quad (210)$$

### 11.1.13 Implementation and Parameter Sets

All equations above are combined in the nonlinear model approach of eq. (211) with the vector of the static variables  $z$ , the state vector  $x$ , the input vector  $u$ , the output vector  $y$ , and the parameter vector  $\theta$ . The 60 static variables  $z$  (containing mass flows, heat flows, etc.) are calculated first and then are used to determine the derivatives  $dx/dt$ . Here, the output vector  $y$  is just a concatenation of the vectors  $x$  and  $z$ .

$$z = h(x, u, \theta) \quad \frac{dx}{dt} = f(x, z, u, \theta) \quad y = [x, z] \quad (211)$$

The state vector  $x$  is defined by eq. (212). Thus, the model is of order 13.

$$x = [L_e, m_{e,v}, p_{e,l}, T_{ov}, T_{cmp}, T_{c,g}, L_c, m_{c,v}, p_{c,l}, T_{sc}, T_{a,o}, T_{w,o}, m_{ice}] \quad (212)$$

The model is implemented in C/C++, using an ODE solver from [MathLib 99]. A variable-order multi-step solver is used, which is especially designed for (moderately) stiff problems (*ode15s*, [Matlab 98]). The set of differential equations is stiff because the time constants are of very different magnitudes. The code is compiled to an executable file, which is called from a MATLAB<sup>®</sup> simulation environment.

For each operating mode, a different parameter vector  $\theta$  is defined (eq. (213)), for considering the different model configurations.

$$\theta = \begin{cases} \theta_0 & (u_{\text{mode}} = 0 = \text{off}) \\ \theta_1 & (u_{\text{mode}} = 1 = \text{on}) \\ \theta_2 & (u_{\text{mode}} = 2 = \text{defrosting}) \end{cases} \quad (213)$$

For normal operation (mode on), all values of the corresponding parameters are given in Appendix C, Section ‘Simulation Model Parameters’. The parameters are either given by the manufacturer, or their physical values are estimated. Some parameters have been adjusted in order to fit the measured data. They are especially marked in Appendix C. In summary, the thermal inertiae  $\tau_i$  of Section 11.1.12, the heat transfer parameters  $kb_i$ , as well as some characteristics of the compressor and of the expansion valve have been adjusted. For adjusting the parameters, the measured data from the residential building at Barzheim were used [Reiner, Shafai 98]. For each operating mode, one parameter set is obtained.

## 11.2 Simulations

After the validation for the nominal case, the heat pump model is coupled with a building model and used to simulate the fault cases.

### 11.2.1 Validation for the Nominal Case

The heat pump simulation model has been validated by comparing it with the measured data from the residential building at Barzheim. The same parameter set (one for each operating mode) is used for simulating all data files of the heating period in the year 1998. The following plots show the results of one day of that heating period (January 29, 1998).

The mode  $u_{mode}$  in Fig. 85 is an input from the heat pump controller, which specifies the three states of the heat pump (off, on, defrosting). During the day selected, the heat pump was switched on three times, whereas the defrosting cycles were started at constant time intervals when the heat pump was running. The heat pump controller itself is not implemented in the simulation model.

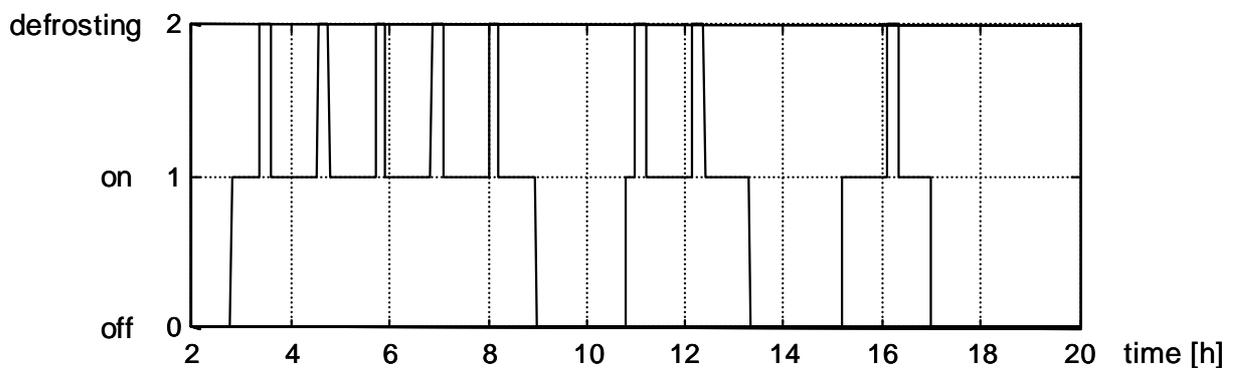


Fig. 85: Operation mode  $u_{mode}$ ,  $0 = \text{off}$ ,  $1 = \text{on}$ ,  $2 = \text{defrosting}$ .

The temperature signals  $T_2..T_{10}$  and the pressures signals  $p_{lp}$ ,  $p_{hp}$  are acquired by the sensors shown in Fig. 84. The signals measured by the sensors do not exactly correspond to the signals in the model, because of the different locations of the sensors. In addition, the time constants of the sensors are not modelled.

Fig. 86 and Fig. 87 show the simulated evaporation pressure  $p_{e,v}$  and condensation pressure  $p_{c,v}$ , compared to the measured low pressure  $p_{lp}$  and high pressure  $p_{hp}$ . Especially in the high pressure signal  $p_{c,v}$ , a superposition of the slow daily transient and the fast switching transients can be observed. Generally, the pressures react very fast when the heat pump is switched on. After a

fast response immediately at shut-down, the high-pressure signal  $p_{c,v}$  and the low pressure signal  $p_{e,v}$  equalize slowly. During the defrosting cycles, a level change as a result of the hot-gas valve switching can be observed.

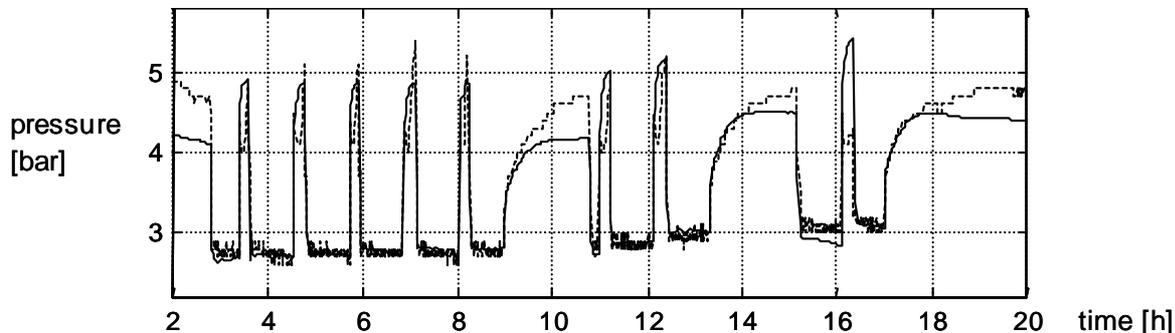


Fig. 86: Simulated evaporation vapour pressure  $p_{e,v}$  (solid line) and measured low pressure  $p_{lp}$  (dashed)

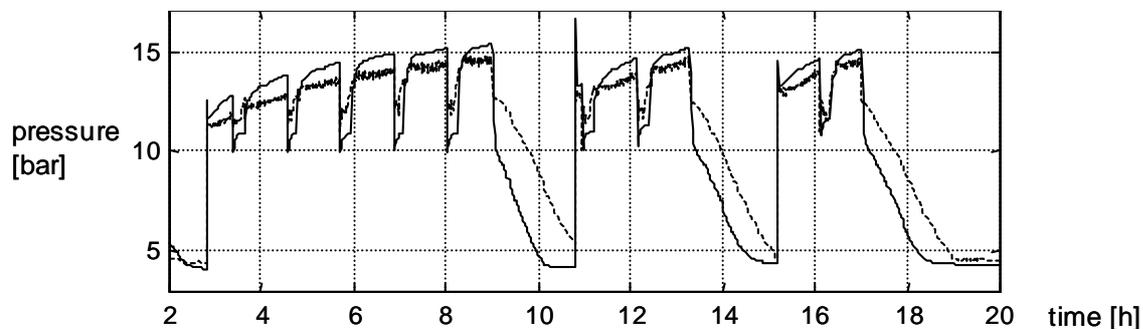


Fig. 87: Simulated condensation vapour pressure  $p_{c,v}$  (solid line) and measured high pressure  $p_{hp}$  (dashed)

In the measured superheating temperature  $T_{ov}$  (Fig. 88), some oscillations are present due to the hunting effect of the expansion valve (cf. Section 4.1). These effects are not modelled.

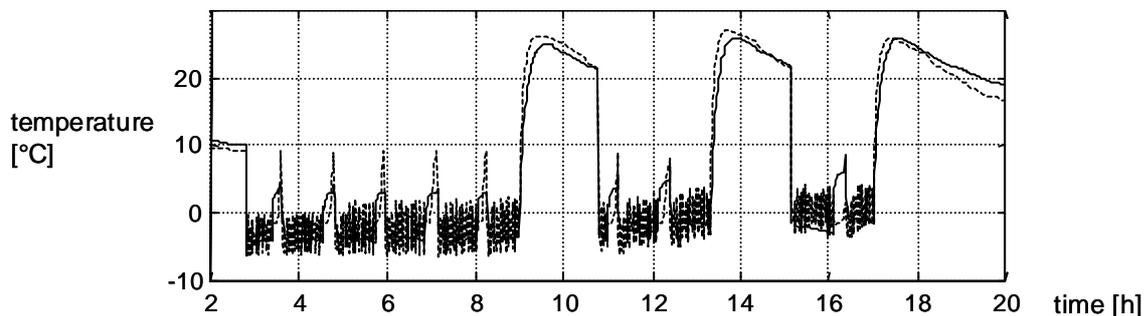


Fig. 88: Simulated superheating temperature  $T_{ov}$  (solid line) and measured temperature  $T_3$  (dashed)

In Fig. 89, the slow reaction of the hot gas temperature  $T_{hg}$  can be seen.

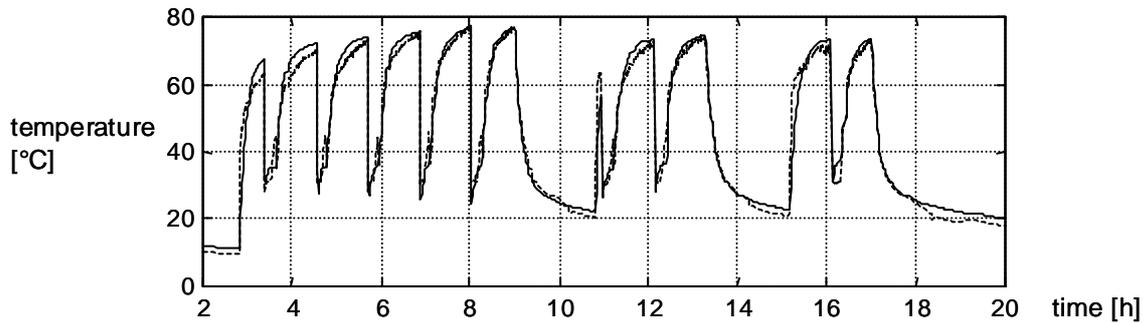


Fig. 89: Simulated hot gas temperature  $T_{hg}$  (solid line) and measured temperature  $T_4$  (dashed)

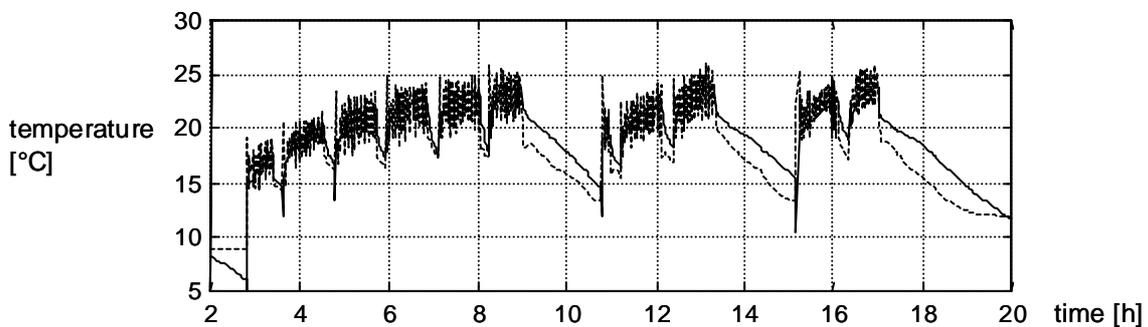


Fig. 90: Simulated subcooling temperature  $T_{sc}$  (solid line) and measured temperature  $T_6$  (dashed)

Fig. 91 shows the water inlet temperature  $T_{w,i}$ , which is an input to the model. The water output temperature  $T_{w,o}$  is shown in Fig. 92. In the water temperatures, the superposition of the slow and fast transients can be observed again. The slow transients of the input temperature  $T_{w,i}$  is a result of the large time constant of the building (cf. Section 11.2.2).

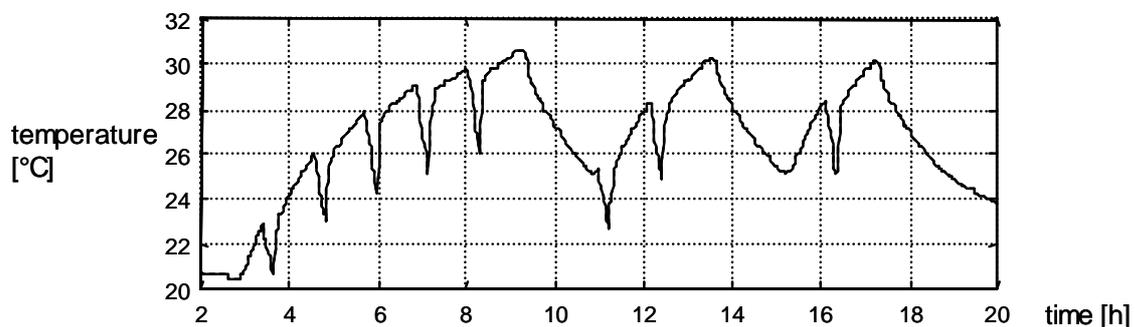


Fig. 91: Measured water input temperature  $T_{w,i}=T_{10}$

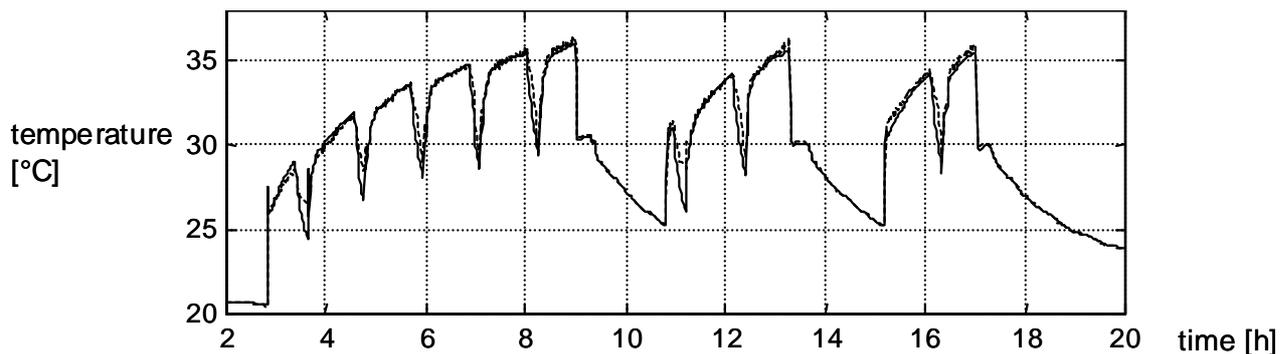


Fig. 92: Simulated water output temperature  $T_{w,o}$  (solid line) and measured temperature  $T_8$  (dashed)

From the internal model signals, the filling levels of the evaporator and the condenser are plotted here. The evaporation length  $L_e$  is plotted in Fig. 93. During the shut-down periods, the evaporation length increases because of the migration process.

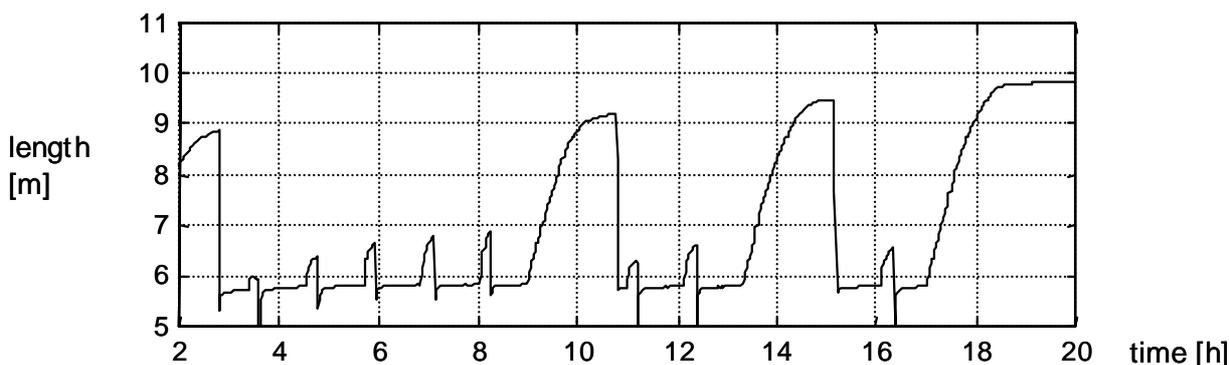


Fig. 93: Simulated evaporation length  $L_e$

During the shut-down periods, the liquid level  $L_c$  in the condenser decreases and may become zero when the condenser is “empty” (cf. Fig. 94).

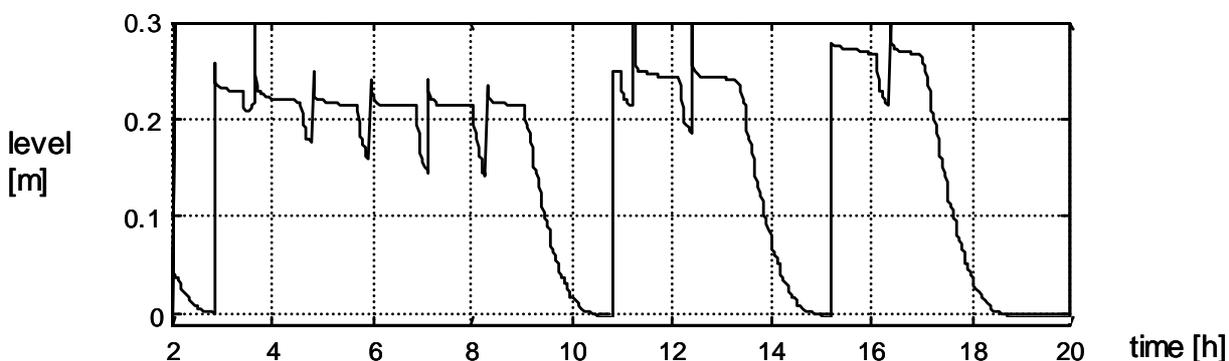


Fig. 94: Simulated liquid level  $L_c$  in the condenser.

From the internal mass flows, the evaporation mass flow  $m_e^*$ , a fast signal, is shown in Fig. 95. The increase of the mass flow during the defrosting se-

quences is a consequence of the changing flow resistance in the system, when the hot gas bypass is open.

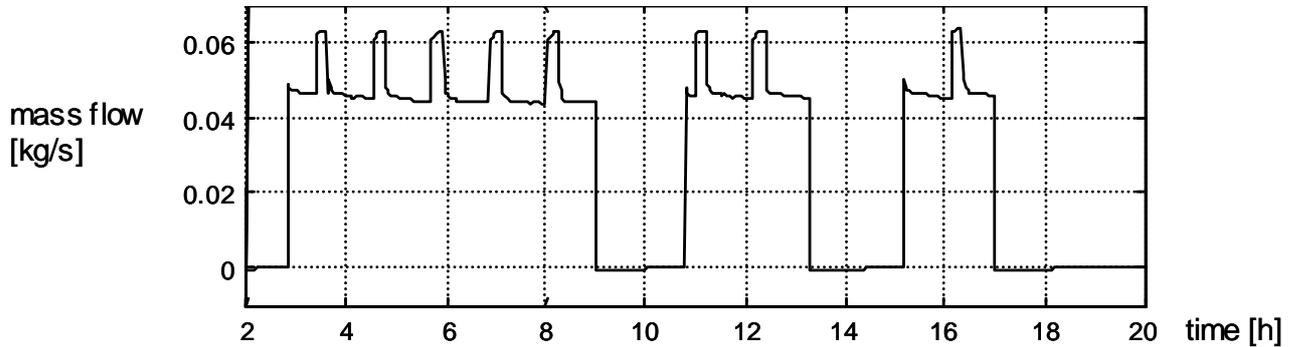


Fig. 95: Simulated evaporation mass flow  $m_e^*$

Due to the icing effect on the evaporator, which is also included in the model, the pressure drop  $\Delta p_a$  on the air side increases (cf. Fig. 96).

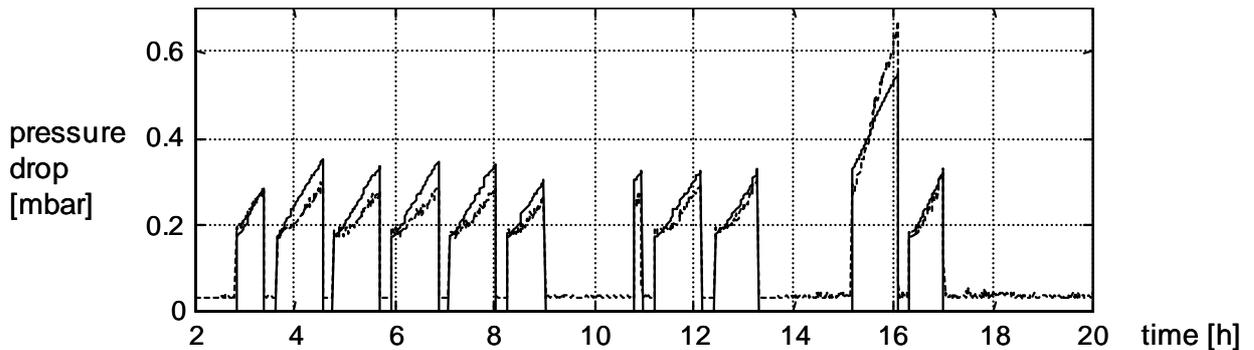


Fig. 96: Simulated (solid line) and measured (dashed) icing pressure drop  $\Delta p_a$

Clearly, the model is able to represent the significant characteristics and dynamics of a heat pump and can be used as a data source for testing fault diagnosis systems.

### 11.2.2 Coupling with Building

For a realistic simulation of the fault cases, the heat pump simulation model is coupled with a building model (cf. Fig. 6 on page 40). A linear one-zone building model of third order from [Reiner, Shafai 98] is used (eq. (214)), with the heating water return flow temperature  $T_{w,i}$  (equal to the water input temperature of the heat pump), the heated floor temperature  $T_f$ , and the room temperature  $T_r$  as state variables. The parameters are identified and averaged for the heating period 1997/98 at Barzheim (validation in [Shafai 99]). They include the heat transfer parameters  $kA_{w,f}$  from the heating water to the floor,  $kA_{f,r}$  from the floor to the room, and  $kA_{r,amb}$  from the room to the environment. Other parameters are the total volume  $V_w$  of the heating water, the mass

$m_f$  of the floor heating system, and the time constant  $\tau_r$  of the building. Inputs to the building model are the environmental temperature  $T_{amb}$  and the heat flow  $Q_{HP}^*$  supplied by the heat pump to the heating water. The values of the building model parameters are also given in Appendix C, Section ‘Simulation Model Parameters’, ‘Building Model’.

$$\begin{aligned}
 \begin{bmatrix} \dot{T}_{w,i} \\ \dot{T}_f \\ \dot{T}_r \end{bmatrix} &= \begin{bmatrix} -\frac{kA_{w,f}}{\rho_w c_w V_w} & \frac{kA_{w,f}}{\rho_w c_w V_w} & 0 \\ \frac{kA_{w,f}}{m_f c_f} & -\frac{kA_{w,f} + kA_{f,r}}{m_f c_f} & \frac{kA_{f,r}}{m_f c_f} \\ 0 & \frac{kA_{f,r}}{\tau_r kA_{r,amb}} & -\frac{kA_{f,r}}{\tau_r kA_{r,amb}} - \frac{1}{\tau_r} \end{bmatrix} \cdot \begin{bmatrix} T_{w,i} \\ T_f \\ T_r \end{bmatrix} \\
 &+ \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\tau_r} \end{bmatrix} \cdot T_{amb} + \begin{bmatrix} \frac{1}{\rho_w c_w V_w} \\ 0 \\ 0 \end{bmatrix} \cdot Q_{HP}^*
 \end{aligned} \tag{214}$$

### 11.2.3 Simulations for Fault Cases

With the coupled models, nine fault cases are simulated by changing the appropriate parameters of the heat pump model (cf. Table 19). The input signals, such as the outdoor air temperature  $T_{a,i} = T_{amb}$ , the outdoor air humidity  $\varphi_{a,i}$ , and the controller input  $u_{mode}$ , are extracted from the measured data of the same heating period 1997/98 as above. The parameters are reduced by the factor of eq. (215), with  $f_i \in [0..1]$  being the size of the fault.

$$\theta_{i,f} = \theta_{i,0} \cdot (1 - f_i) \tag{215}$$

All faults are listed in Table 19 with their corresponding model parameters (cf. Table 6 on page 105). For simulating the evaporator fault 2, the heat transfer parameters  $kb_{e,e}$  and  $kb_{e,ov}$  are reduced (cf. Sections 11.1.1 and 11.1.2). In the real heat pump, this fault corresponds to evaporator fouling. The air mass flow  $m_{a,0}^*$  (fault 3) and the water mass flow  $m_{w,0}^*$  (fault 5) are defined to be constant during the simulation and therefore are regarded as parameters. For simulating refrigerant leakage (fault 9), the initial conditions of the state variables  $L_{e,0}$ ,  $L_{c,0}$ , (evaporation length and condensation level) as

well as  $m_{e,v,0}$ ,  $m_{c,v,0}$  (vapour mass of evaporator and condenser) are changed. They influence the filling of the refrigerant cycle.

Table 19: The parameters of the simulation model. The faults are simulated by changing the corresponding parameters.

fault	simulation model parameters (cf. Sections)	fault description (faults in real applications)
fault 1	-	normal behaviour, no fault
fault 2	$kb_{e,e}$ , $kb_{e,ov}$ (11.1.1, 11.1.2)	reduced evaporator heat transfer (evaporator fouling)
fault 3	$m_{a,0}^*$ (11.1.8)	reduced evaporator air mass flow (evaporator/air channel fouling, air fan malfunction)
fault 4	$kb_{c,c}$ , $kb_{c,g}$ (11.1.4, 11.1.5)	reduced condenser heat transfer (condenser fouling)
fault 5	$m_{w,0}^*$ (11.1.9)	reduced condenser water mass flow (fouling, water pump malfunction)
fault 6	$kb_{he,e}$ , $kb_{he,ov}$ (11.1.1, 11.1.2)	reduced internal heat exchanger heat transfer (internal heat exchanger fouling)
fault 7	$\lambda_{cmp}$ (11.1.3)	reduced compressor efficiency (compressor malfunction)
fault 8	$k_{exp}$ (11.1.7)	reduced expansion valve flow rate (expansion valve malfunction)
fault 9	$L_{e,0}$ , $m_{e,v,0}$ , $L_{c,0}$ , $m_{c,v,0}$ (11.1.1, 11.1.5)	reduced refrigerant charge (refrigerant leakage)

Both evaporator fouling (fault 2) and evaporator icing may be present at the same time. Then, the heat transfer parameters  $kb_{e,e}$  and  $kb_{e,ov}$  are influenced by fault 2, whereas the heat transfer reduction factor  $k_{ice}$  is influenced by the icing effect (eqs. (182) and (183)). However, the effect on the signals is the same for both cases.

Fig. 97 through Fig. 99 show the simulation results for fault 2 of the size  $f_2 = 0.2$  (eq. (215)), compared to the nominal case (data of January 29, 1998). Only the signals with the largest deviations from the nominal case are plotted. By reducing the heat transfer rate in the evaporator, the evaporation pres-

sure  $p_{e,v}$  and the corresponding dew-point temperature  $T_d(p_{e,l})$  decrease ( $p_{e,l} \approx p_{e,v}$ ). Thus, the superheating temperature  $T_{ov}$  is also at a lower level.

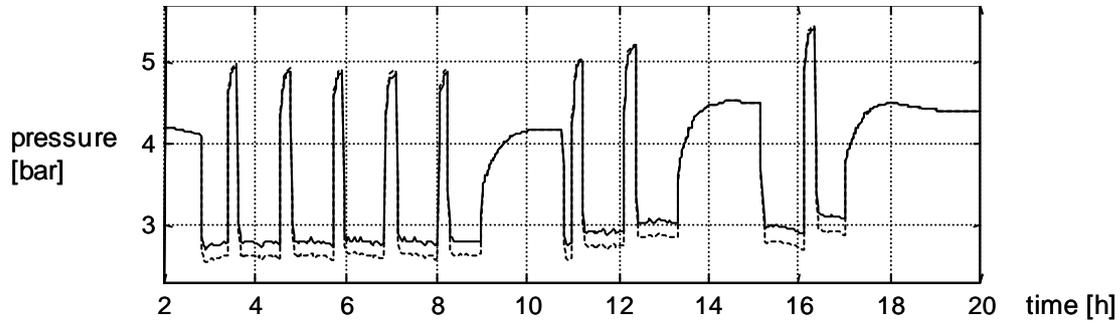


Fig. 97: Evaporation vapour pressure  $p_{e,v}$ , simulation for fault 2 (dashed) and nominal case (solid).

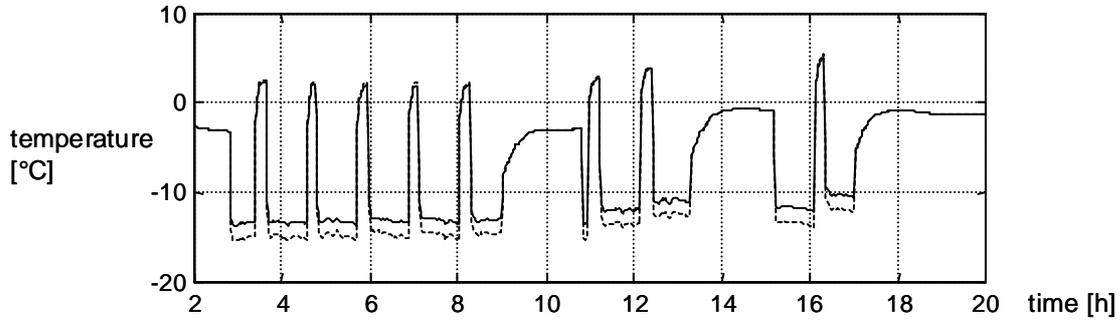


Fig. 98: Evaporation dew-point temperature  $T_d(p_{e,l})$ , simulation for fault 2 (dashed) and nominal case (solid).

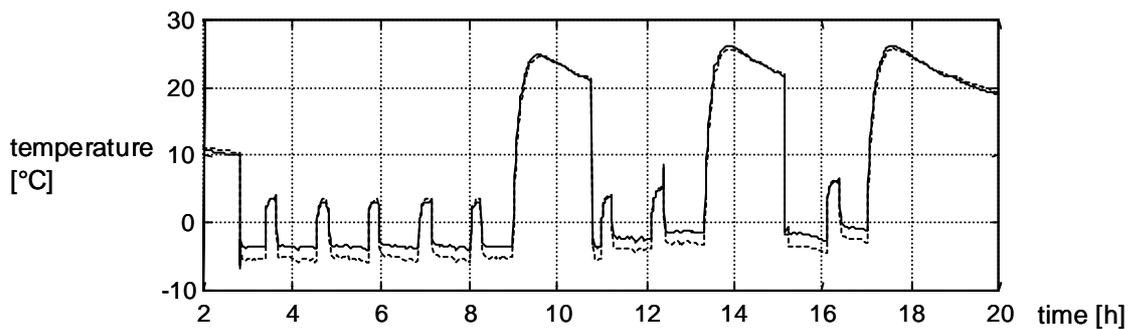


Fig. 99: Superheating temperature  $T_{ov}$ , simulation for fault 2 (dashed) and nominal case (solid).

All fault cases are simulated with the same sizes  $f_i = 0.2$  (with one exception for fault 9,  $f_9 = 0.1$ , see below) and the significantly influenced temperature and pressure signals are summarized in Table 20 (signals in steady state). The first row contains the results for fault 2 from above. A reduction of the air mass flow (fault 3) has similar effects on the evaporation pressure and temperatures, but here the air outlet temperature  $T_{a,o}$  decreases intensively due to a higher temperature drop. In the case of fault 4, only the gas temperature  $T_{c,g}$  increases significantly. With a larger fault ( $f_4 \gg 0.2$ ), the result is analogous to that of fault 2. By reducing the water mass flow through the condenser (fault 5), the condenser pressure ( $p_{c,l}, p_{c,v}$ ) and the temperatures ( $T_{b,m,d}(p_{c,l})$ ) increase, which is in analogy to fault 3. Regarding fault 6, the subcooling temperature  $T_{sc}$  increases with a reduced heat transfer in the internal heat exchanger. Since the superheating temperature  $T_{ov}$  is controlled by the expansion valve, it is not significantly influenced by fault 6. With fault 7, the compressor mass flow is reduced, which obviously results in a higher evaporation pressure ( $p_{e,l}, p_{e,v}$ ) and a lower condensation pressure ( $p_{c,l}, p_{c,v}$ ). The controller of the expansion valve tends to compensate a reduction of the expansion valve mass flow (fault 8) by opening the valve ( $u_{exp} \uparrow\uparrow$ ). Because of the steady-state error of the P-controller, the superheating temperature  $T_{ov}$  is increased slightly. The effects of a changing refrigerant filling (fault 9) are shown in the last row.

Table 20: Sensitivities of the temperature and pressure signals to the faults (steady state).

fault	Significantly influenced signals, direction $\downarrow, \uparrow$ and magnitude $((\uparrow)), (\uparrow), \uparrow, \uparrow\uparrow$
fault 2	$p_{e,l}, p_{e,v} \downarrow\downarrow, T_{b,m,d}(p_{e,l}) \downarrow\downarrow, T_{ov} \downarrow\downarrow$
fault 3	$p_{e,l}, p_{e,v} \downarrow, T_{b,m,d}(p_{e,l}) \downarrow, T_{ov} \downarrow, T_{a,o} \downarrow\downarrow$
fault 4	$T_{c,g} \uparrow$
fault 5	$p_{c,l}, p_{c,v} \uparrow, T_{b,m,d}(p_{c,l}) \uparrow, T_{hg} \uparrow, T_{c,g} \uparrow, T_{sc} \uparrow, T_{w,o} \uparrow\uparrow$
fault 6	$T_{sc} \uparrow$
fault 7	$p_{e,l}, p_{e,v} \uparrow\uparrow, T_{b,m,d}(p_{e,l}) \uparrow\uparrow, p_{c,l}, p_{c,v} \downarrow\downarrow, T_{b,m,d}(p_{c,l}) \downarrow\downarrow,$ $T_{ov} \uparrow, T_{hg} \downarrow\downarrow, T_{c,g} \downarrow\downarrow, T_{sc} \downarrow\downarrow, T_{a,o} \uparrow, T_{w,o} \downarrow\downarrow$
fault 8	$T_{ov} (\uparrow), T_{hg} ((\uparrow)), u_{exp} \uparrow\uparrow$
fault 9	$p_{c,l}, p_{c,v} \uparrow\uparrow, T_{b,m,d}(p_{c,l}) \uparrow\uparrow, T_{hg} \uparrow, T_{c,g} \downarrow, T_{sc} \uparrow\uparrow$

In the next step, various fault sizes are simulated for all fault cases of Table 19. For the faults 2..8, four sizes are simulated:

- fault size 2:  $f_i = 0.2$  (20% parameter change)
- fault size 3:  $f_i = 0.3$  (30% parameter change)
- fault size 4:  $f_i = 0.4$  (40% parameter change)
- fault size 5:  $f_i = 0.5$  (50% parameter change)

For fault 9, the three simulated sizes are smaller, since its effect on the signals is larger:

- fault size 1:  $f_i = 0.1$  (10% parameter change)
- fault size 2:  $f_i = 0.2$  (20% parameter change)
- fault size 3:  $f_i = 0.3$  (30% parameter change)

For the training of the fault diagnosis system described in Section 6.2, some ‘virtual’ sensors ( $T_1..T_{10}$ ) are defined by the simulated signals of eqs. (216) to (219) (cf. Fig. 40 on page 104 and Fig. 84 on page 174).

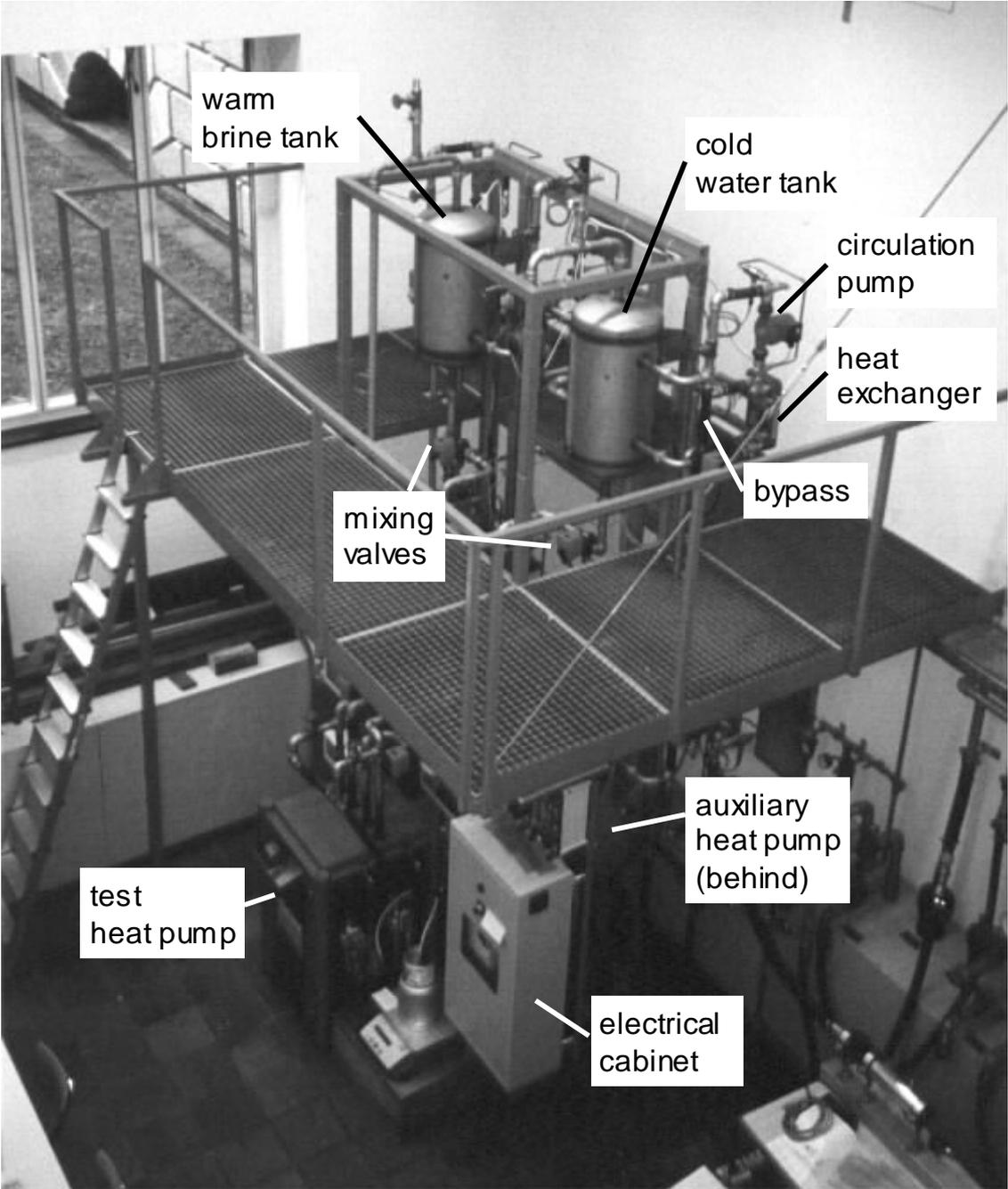
$$T_1 = T_b(p_{e,l}) \quad T_2 = T_d(p_{e,l}) \quad T_3 = T_{ov} \quad (216)$$

$$T_4 = T_{hg} \quad T_5 = T_b(p_{c,l}) \quad T_6 = T_{sc} \quad (217)$$

$$T_7 = T_{a,o} \quad T_8 = T_{w,o} \quad (218)$$

$$T_9 = T_{a,i} \quad T_{10} = T_{w,i} \quad (219)$$

# 12 IMRT Test Bench



*Fig. 100: The test bench at the Measurement and Control Laboratory, IMRT (seen from above).*

A new dynamic test bench for commercial brine-to-water heat pumps has been installed (Fig. 100). It is used for training and testing of the fault diagnosis systems. Therefore, the commercial test heat pump from SATAG<sup>®</sup> Thermotechnik AG has been modified in order to generate data for different fault cases. A full automation of the test cycles has been realized, which allows to acquire the data sets without any manual intervention. The test bench is especially designed for an emulation of the dynamic behaviour of the heat source side (ground pipe cycle) as well as the heat sink side (heating water cycle and building).

Compared to testing in a real building (like that at Barzheim), the advantages of a test bench are obvious:

- There are no seasonal restrictions due to the heating periods.
- The results are reproducible, since there are no unknown influences from the weather or the inhabitants.
- The fault cases can be tested without any restrictions for the inhabitants.
- The diagnosis system *FuzzyWatch* is systematically and automatically trained within a short time (a few days up to one week).
- The test bench can be used for testing heat pump controllers or new heat pump types, as well.

The concept of the test bench was worked out in a previous step, whereas the test bench has been built within the scope of this thesis.

## 12.1 Test Environment

A brine cycle is realized on the heat source side and a water cycle is realized on the heat sink side (Fig. 101). The brine cycle (on the left side) contains a warm storage tank (on the top) and a cold storage tank (on the bottom). The outputs of both tanks are mixed in order to get the desired temperature at the input of the test heat pump (mixing valve between the tanks). The upper and lower tank temperatures are controlled by activating the two heat exchangers (on the left side) by opening or closing the corresponding bypass valves (between the tanks and the heat exchangers). The heat exchangers withdraw the heat from the warm laboratory medium, which operates at about 30°C.

The water cycle works in the same way as the brine cycle, where two heat exchangers dispense the heat to the cold laboratory medium, which operates at about 14°C (on the right side). An additional heat exchanger between the water cycle and the brine cycle allows the recycling of the heat from the warm side to the cold side (on the top, between the tanks). Thus, a part of the heating energy can be saved. It is also desirable to control the temperatures when the test heat pump is switched off. Therefore, an auxiliary heat pump is installed, which permits to increase the difference between the brine and the water temperature levels additionally.

For controlling the flows, 3-way valves and 2-way valves are used, whereas numerous pumps induce circulation. The signals are measured by four types of sensors, such as absolute Pt100 temperature sensors (T), temperature difference sensors at the heat exchangers ( $dT$ , combinations of two thermocouples), pressure difference sensors at the circulation pumps ( $dp$ , membrane transmitters), and flow meters (F, magnetic-inductive) at the outputs of the test heat pump. From the pressure differences, the volumetric flows are estimated on the basis of the characteristics of the pumps, and from the temperature differences, the heat flows are calculated.

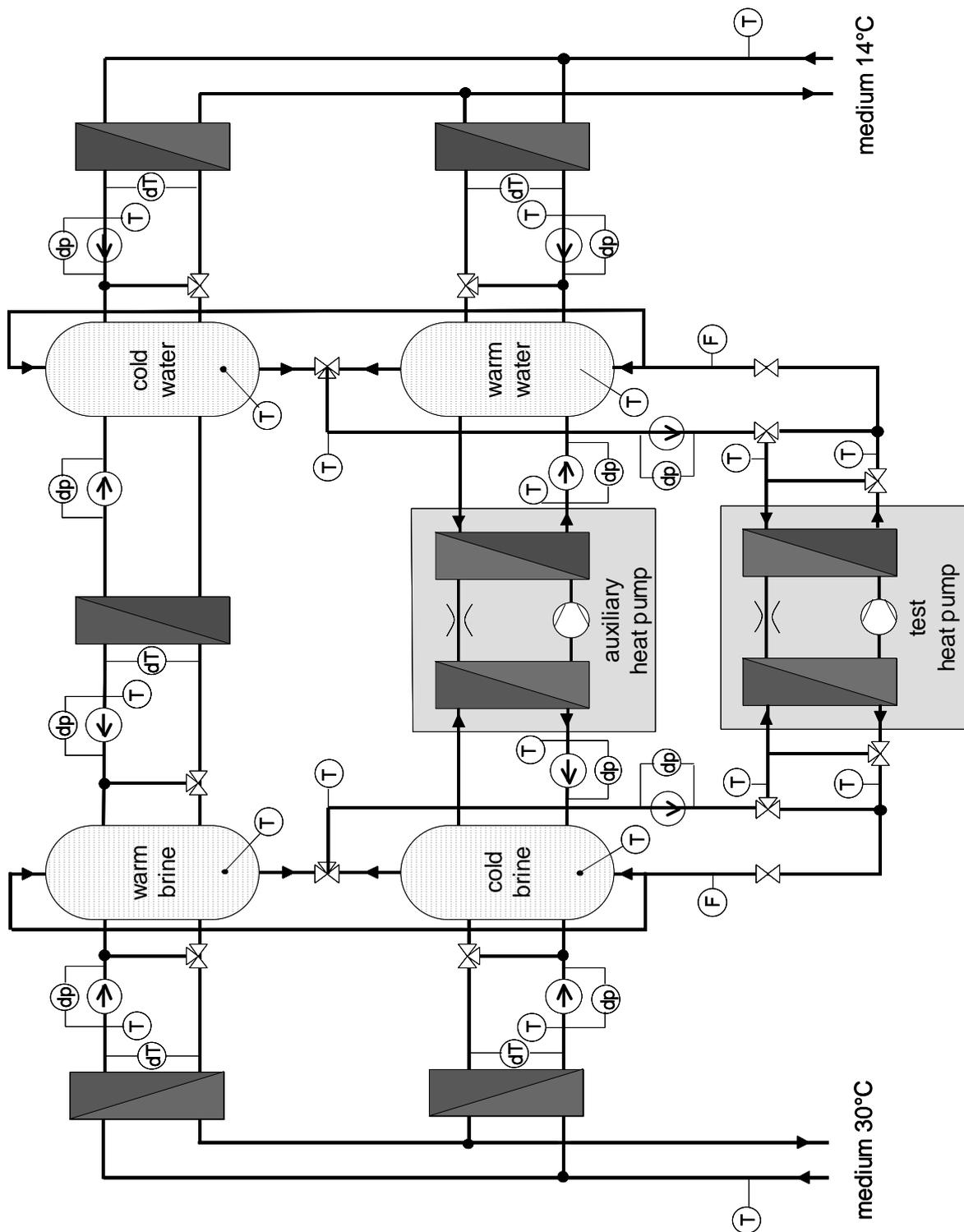


Fig. 101: The test environment with two brine tanks, two water tanks, five heat exchangers, an auxiliary heat pump and the test heat pump.  $T$  = temperature sensors,  $dT$  = temperature difference sensors,  $dp$  = pressure difference sensors,  $F$  = flow sensors.

## 12.2 Data Acquisition and Automation

A digital bus system by *WAGO*<sup>®</sup> is implemented (Fig. 102). Its five field nodes contain input modules with integrated A/D converters and signal conditioning as well as output modules with integrated D/A converters or relays. These nodes are connected to three PCs by a *CAN* bus. Two PCs are used as realtime target systems for the *MATLAB*<sup>®</sup> *xPC Target*<sup>®</sup> software, while one PC is used for the *LabView*<sup>®</sup> software. The *CAN* master card controls the bus system and is also plugged into one of the targets. Both of the targets are connected to one host by *TCP/IP*, which is controlling the targets. The PCs are connected to the intranet for exchanging data.

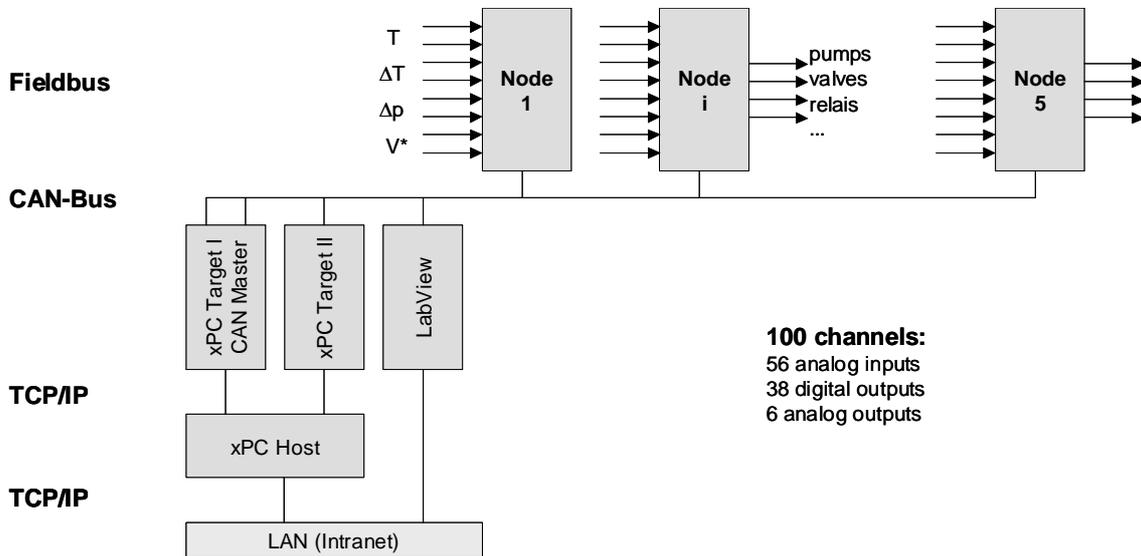


Fig. 102: The bus system of the test bench.

The tasks for the operation of the test bench are assigned to the various PCs. Target I controls the test environment and optionally runs the emulation, whereas with Target II, the fault diagnosis systems are tested. For other projects, heat pump controllers can be tested, as well. *LabView*<sup>®</sup> is used for data visualization and manual control of the test bench.

## 12.3 Running Test Cycles

### 12.3.1 Definition of the Test Cycles

For all fault cases of the test heat pump, several test cycles with on-off switching are run. For this purpose, a special software tool has been developed. A sample test cycle is illustrated in Fig. 103. Such a test cycle is repeated for each fault case.

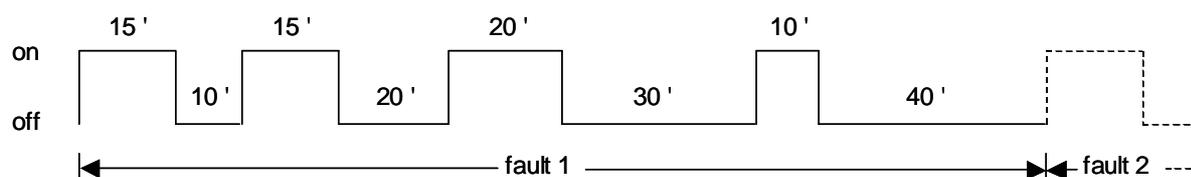


Fig. 103: A sample test cycle.

The duration of the on sequences (normal operation) are chosen to be shorter than in real applications, since for the training of the fault diagnosis systems, only the first 10 minutes of each sequence are relevant, until a (quasi) steady state is reached. For the test cycles, the duration is varied between 10 and 20 minutes for the on sequences. The duration of the off sequences is varied more in order to vary the initial conditions for the subsequent on sequences. During the off sequences the high and low pressure levels (and temperature levels) are equalized slowly by the migration process in the heat pump. Thus, the initial conditions are at different pressure levels, depending on the duration of the preceding off sequence. With this method, the transients at the beginning of the on sequences can be varied for each sequence, which is more realistic.

The total duration for the training of the diagnosis system *FuzzyWatch* is estimated for the following assumptions (cf. applications of Chapter 6):

- The duration of the on sequences is 10 minutes and the average duration of the off sequences is 20 minutes, which adds up to 30 minutes per data point (= parameter set). Usually the on sequences are evaluated only.
- 10 data sequences per fault case are acquired (10 data points per cluster).
- 10 fault cases are trained (10 fault clusters, nominal case included).

This training lasts  $30 \times 10 \times 10$  minutes, which is 50 hours or about 2 days. If more fault cases have to be trained (for example the most common fault combinations), or if more different operating conditions are covered, then the duration might be longer, up to about one week.

### 12.3.2 Operating Conditions

The valve positions of the test environment (Fig. 101) are kept constant during one test cycle. By varying the positions between the test cycles, the operating conditions are varied as well:

**Heat transfer at the heat source side.** With the 3-way valves on the left of the upper and lower brine tanks, the heat transfer from the warm medium to the brine cycle is influenced, which refers to the heat transfer from the ground to the brine fluid in real ground pipes.

**Heat transfer at the heat sink side.** With the 3-way valves on the right of the upper and lower water tanks, the heat transfer from the water cycle to the cold medium is influenced, which refers to the heat transfer from the water to the room in the heating system of a real building.

**Inertia of the heat source side.** With the mixing 3-way valve between the warm brine tank and the cold brine tank, the total volume of the brine cycle is changed by either activating one tank only (upper or lower position) or both tanks (mid position). In reality, the heat source ‘inertia’ refers to the type of the ground cycle and its operating conditions.

**Inertia of the heat sink side.** In the same way, the ‘inertia’ of the water cycle is influenced by the mixing valve between the cold water tank and the warm water tank. In reality, the heat sink ‘inertia’ refers to the type of house and its operating conditions.

### 12.3.3 Emulation

In order to test the heat pump under more realistic conditions, an emulation of the heat source side as well as the heat sink side will be implemented in the future. Therefore, a model of the ground cycle and a model of the building will run in parallel to the real heat pump (Fig. 104). In realtime, these models will calculate the setpoints for the heat pump input temperatures (brine temperature  $T_{br,return,sp}$ , water temperature  $T_{w,return,sp}$ ) as well as the setpoints for the volumetric flows through the heat pump (brine  $V_{br,sp}^*$ , water  $V_{w,sp}^*$ ) based on the measured heat pump output temperatures ( $T_{br,supply}$ ,  $T_{w,supply}$ ) and on stored weather data (ground temperature  $T_{ground}$ , ambient temperature  $T_{amb}$ , solar radiation  $Q_{radiation}^*$ , etc.). Four controllers are fed by the setpoints. The outputs of these four controllers are the inputs for a hierarchical control system of the test bench. Primarily they influence the positions of the mixing valves between the upper and lower tanks for the temperature control as well as the valves for flow control (cf. Fig. 101 on page 208).

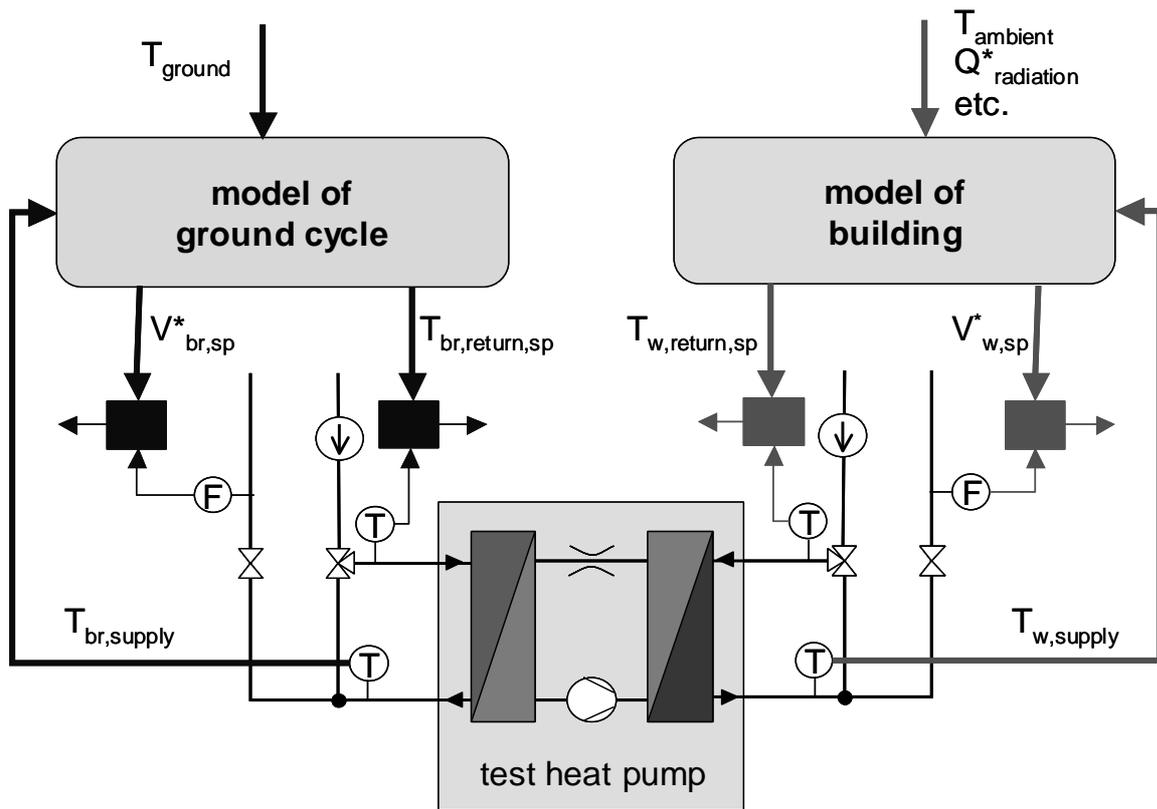


Fig. 104: Emulation of the ground cycle and the building.

## 13 Summary and Conclusions

The diagnosis systems *FuzzyWatch* and *HeatWatch* have been tested and validated for several applications. Two software tools have been developed with graphical user interfaces, which allow to visualize the diagnosis results, and which allow to train or initialize the diagnosis systems in a convenient way. For both diagnosis systems, the heat pump or the refrigerating machine is assumed to be switched on and off by the controller, which produces typical data sequences, characterized by step responses.

***FuzzyWatch.*** In a first step, the parameters of a gray-box model are re-identified for each data sequence, and in a second step, the faults are classified. With this diagnosis system, single faults, gradual faults, or even simultaneous faults are classified with a low rate of wrong classifications. With a small number of sensors, a large number of fault cases is classified (cf. Table 21).

*Table 21: FuzzyWatch - number of required sensors, number of classified fault cases, and classification quality (some selected results)*

	required sensors (minimum)	classified faults	classification quality for single faults (one size)
Application II: ZHW test bench	4..5 temperature sensors	1+6 fault cases	<i>training data, standard clusters</i> (cf. Table 5): 4..8% wrong classifications without averaging
Application III: Simulation data	6 temperature sensors	1+8 fault cases	<i>validation data, standard clusters</i> (cf. Table 7): 9% wrong classifications without averaging
Application IV: IMRT test bench	4..5 temperature sensors (4 of which are standard)	1+4 fault cases	<i>validation data, standard clusters</i> (cf. Table 11): 6..10% wrong classifications without averaging <i>vector clusters</i> (cf. Table 12): 1..4% wrong classifications without averaging 0% wrong classifications with averaging (fault sizes 20%, 30%) 0% false alarms

The higher the number of sensors that are chosen for one application, the higher the classification quality is. By averaging the identified parameters over several data sequences before classification, the rate of wrong classifications is significantly reduced (with the tested data down to 0%). The false alarm rate is also minimized by appropriate methods (nominal weighting for standard clusters or minimal fault grade for vector clusters).

The diagnosis system needs to be trained with data for the most common fault cases (or fault combinations). The data for the fault cases has to be acquired on a test bench (as in applications II and IV) or it has to be generated by a simulation model (as in application III). The simulation model first has to be validated with nominal data (which is done in Section 11.2.1). On a test bench, the training data acquisition takes a few days up to one week (cf. Section 12.3.1).

With the software tool developed, the modeling step is automated. After the selection of the (output) signals, the corresponding submodels are defined. The training procedure is fully automated (pressing one button).

**HeatWatch.** The parameters of a physical steady-state model are re-identified for each data sequence. The deviations to the reference parameters are then calculated. Since no fault classification is integrated, the faults are interpreted from the parameter deviations by the service technician or by the user. Generally all parameters are influenced in a certain way by one fault case, because of the physical couplings between the parameters. But only one or very few parameters change significantly, which has been shown for the data of the IMRT test bench. Thus, the interpretation of the parameter deviations is straightforward. The number of required sensors is given by the physical equations (cf. Table 22 for the applications I and IV). With this method, a high diagnosis quality is reached. By averaging the identified parameters over several data sequences, the rate of wrong diagnosis results is significantly reduced (with the tested data even to 0% for application IV).

By measuring the electrical power input of the compressor with an energy counter, no mass flow characteristics of the compressor have to be used to calculate the refrigerant mass flow. By reducing the number of sensors, the number of identifiable parameters would be reduced as well.

Apart from the initialization with nominal data, no training is needed on the test bench. A software tool is developed, which contains the physical steady-state equations of the heat pump model. The initialization step can be executed at any time by pressing one button.

Table 22: *HeatWatch* - number of required sensors, number of identified parameters, and diagnosis quality (selected results)

	required sensors (maximum number)	identified parameters, diagnosis quality
Application I: Barzheim data	7 temperature sensors (4 of which are standard) 2 pressure sensors 1 energy counter (el. power)	11 parameters 5+10 characteristics standard deviation within one day: (cf. Fig. 68) ca. 10% for 7 parameters 10%..20% for 3 parameters ca. 20% for 1 parameter
Application IV: IMRT test bench	7..9 temperature sensors (4 of which are standard) 2 pressure sensors 1 energy counter (el. power)	parameters: see above 0% wrong diagnosis results with averaging (fault sizes 20%, 30%) (cf. Table 18) 0% false alarms

**Comparing *FuzzyWatch* and *HeatWatch*.** The highlights of both diagnosis systems are shown in Table 23, which summarizes the sections above [Zogg 00]. For both diagnosis systems, the computational effort for parameter identification is rather low, since either steady-state or ARX models are used. Nevertheless, the parameter identification algorithms can be replaced by much faster approaches (cf. outlook in Chapter 14 and Appendix A). The computational effort for the classification step (in the case of *FuzzyWatch*) is low, because the calculation of the cluster membership grades is a simple function evaluation.

**Simulation Model.** A detailed physical simulation model for an air-to-water heat pump with an internal heat exchanger and a hot gas defrosting cycle has been developed. The highly nonlinear model is of order 13 with a total of 73 output variables, which are simulated. The model is designed for three operating modes (off, on, defrosting), with one parameter set for each mode. After hand-tuning of certain parameters, the model has been validated with nominal data from one heating period of a residential building (Barzheim). It is used for generating training data for the fault diagnosis system *FuzzyWatch*. Nine fault cases have been simulated by changing the corresponding parameters.

Table 23: Highlights of the diagnosis systems

<i>FuzzyWatch</i>	<i>HeatWatch</i>
few sensors required	
displays faults directly (fault classification included)	displays physical parameters, which can be interpreted in a straightforward way
	no fault training is needed on the test bench
modeling and training are automated by the software tool	needs a physical model in steady state, which has to be defined beforehand
low computational effort for online diagnosis (parameter identification and calculation of the cluster membership grades)	low computational effort for online diagnosis (parameter identification in steady state)
allows a fast commercial realization, but with some training effort on a test bench	allows a fast commercial realization

**IMRT Test Bench.** In the Measurement and Control Laboratory, a fully automated test bench has been built for an commercial brine-to-water heat pump. In contrast to the ‘outdoor’ tests in real buildings, here the experimental results are reproducible and the fault cases can be tested without any restrictions. A modified commercial heat pump with additional valves and sensors has been used to test both diagnosis systems.

# 14 Outlook

## 14.1 Suggestions for the Realization Processes

For the realization of the fault diagnosis systems, the effort has to be known that is needed to apply the systems to a specific (new) machine or installation. Therefore, the cases of a new machine type, a scaled up or scaled down machine of the same type, or the commissioning of an individual machine of a known type, must be distinguished.

### 14.1.1 FuzzyWatch

Fig. 105 shows the realization process for FuzzyWatch over time. For a new machine type, one heat pump has to be modified by adding actuators and it has to be equipped with additional sensors. On a test bench, the data of each fault case is measured and the diagnosis system is trained. The model structure is built automatically by defining the required signals. During the training phase, the number of sensors is reduced.

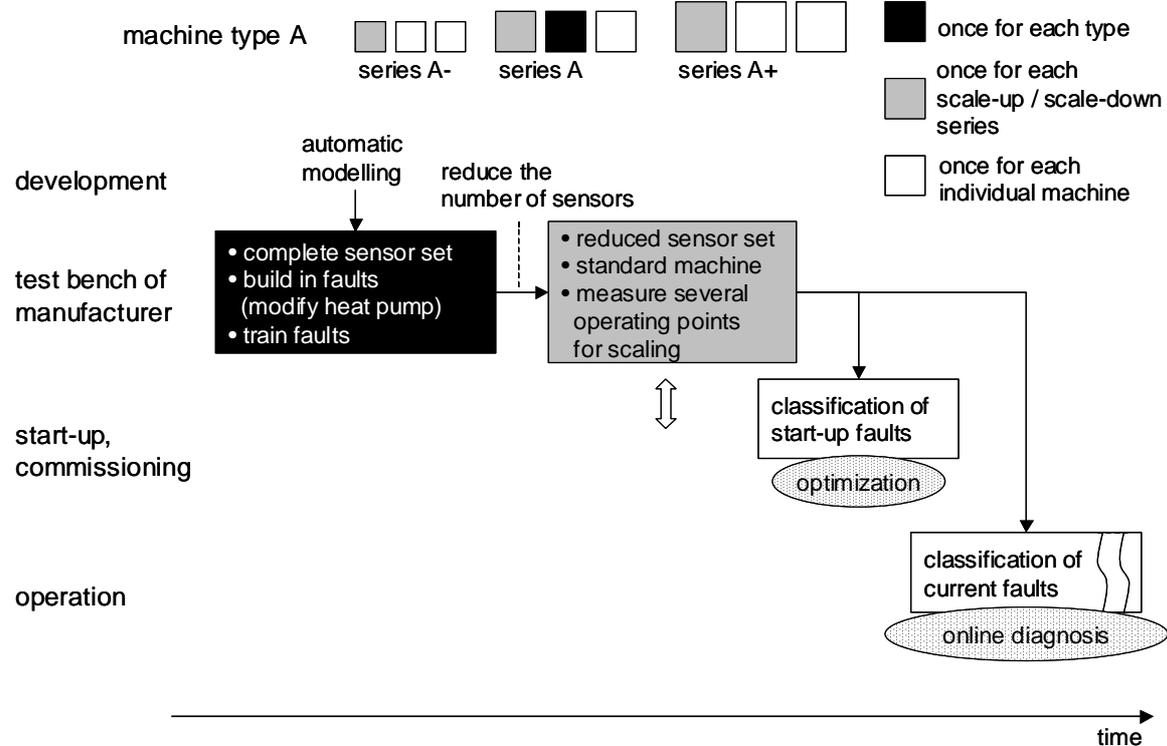


Fig. 105: FuzzyWatch: Realization phases over time.

Once it is trained, the diagnosis system can be applied to all machines of one series (which has to be tested by a portability check). For each scale-up or

scale-down series, the data of one machine has to be measured at one or several operating points. Therefore, no modifications of the machine are required and the reduced sensor set may be used. With this data, the classification system is scaled, which means that the location of the nominal cluster or the origin of the fault vectors is determined (up-/downscale portability). This procedure can also be carried out for each individual machine during commissioning.

At start-up, the faults are classified for each individual machine using the trained diagnosis system. Any faults are analyzed and the system is optimized. During operation, the current faults are classified online. From the sizes of the faults, the condition of the machine can be concluded.

Alternatively, the fault training procedure on the test bench can be replaced by simulations using the detailed simulation model of Chapter 11 (Fig. 106). In this case, the simulation model has to be adapted once for a new machine type (which is a rather high effort), and the new model has to be validated on a test bench. Therefore, just the data from the nominal (fault-free) case is required. Thus, no modifications of the machine are needed, but it has to be equipped with a complete set of temperature and pressure sensors in order to get validation data.

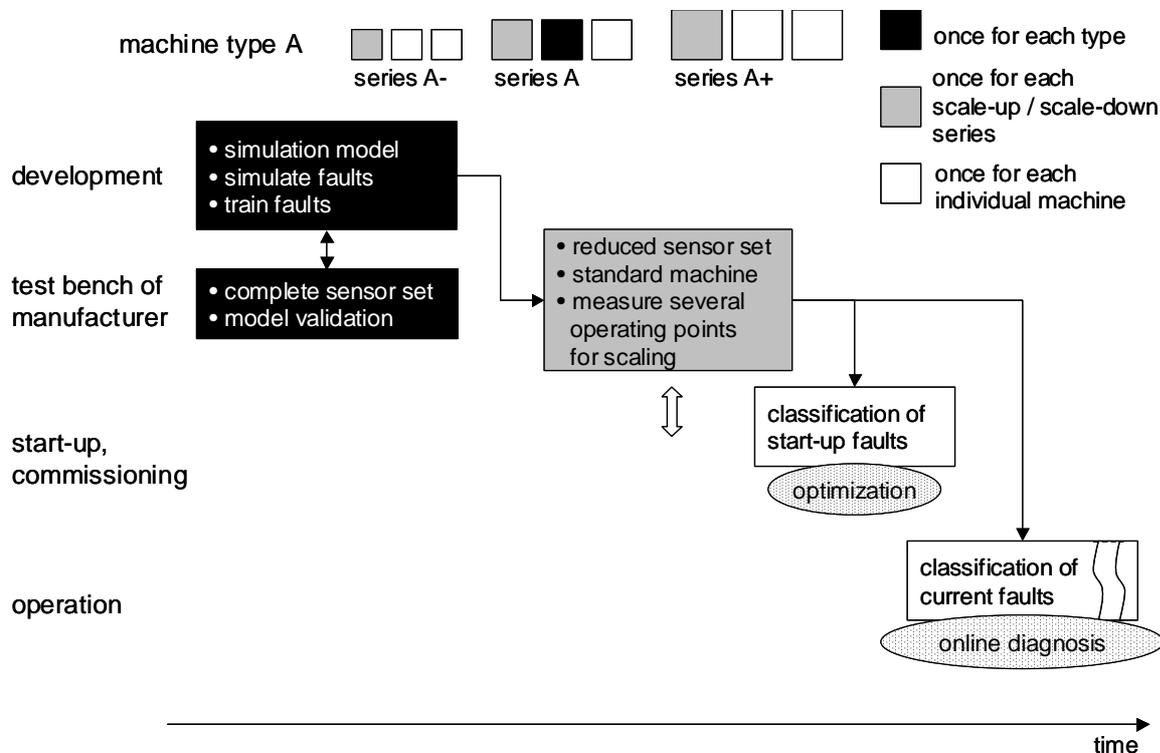


Fig. 106: FuzzyWatch with training by simulation: Realization phases over time.

Using the simulation model, the data for the fault cases is then generated and the diagnosis system is trained. The rest of the realization process is identical to the one shown in Fig. 105.

### 14.1.2 HeatWatch

Fig. 107 shows the realization process for HeatWatch with the realization phases over time. For a new type of a machine, the steady-state model described in Section 8.2 has to be adapted. On the test bench of the manufacturer, the data of one or several operating points are measured and the manufacturer's reference parameters are identified (which may vary from the design parameters). This first initialization step has to be repeated once for each new scale-up or scale-down series (with higher or lower heating power).

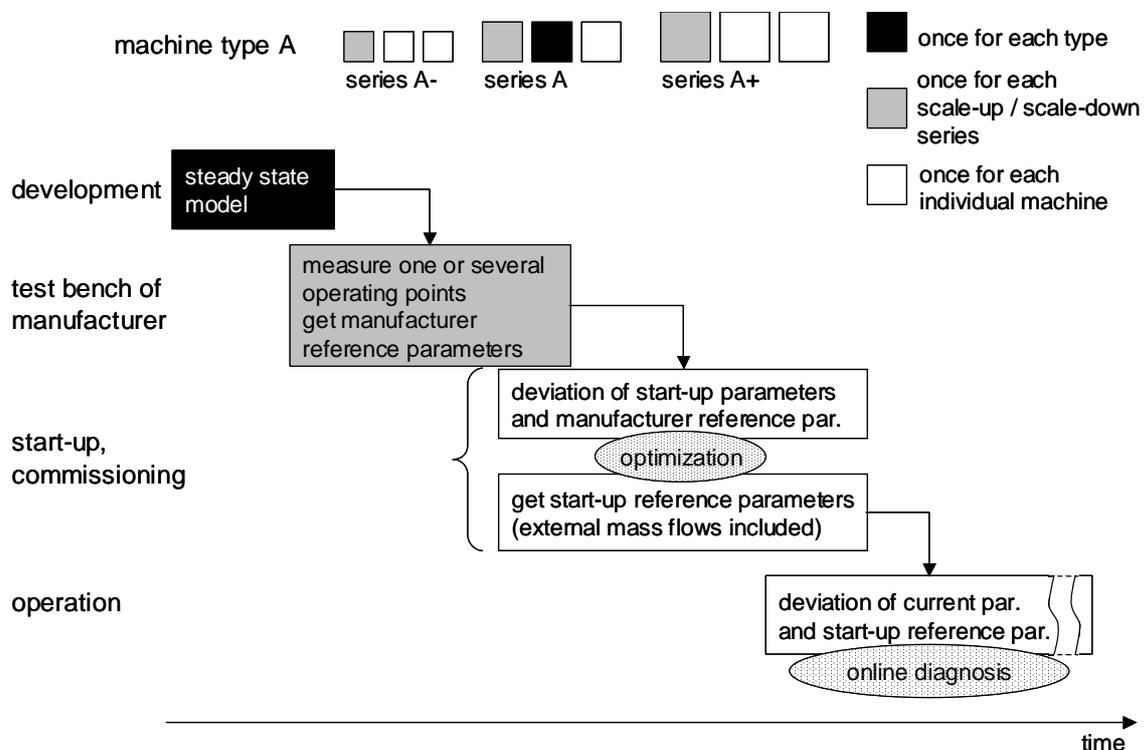


Fig. 107: HeatWatch: Realization phases over time

At start-up, the parameters are identified for each individual machine and compared to the manufacturer's reference parameters. Any parameter deviations are analyzed, and the system is optimized. During the second initialization step, the start-up reference parameters are identified. These parameters are not only influenced by the machine itself, but also by its environment. Especially the external mass flows of a heat pump, such as the brine and water mass flows, vary from one building to another.

During operation, the current parameters are identified online and compared to the start-up reference parameters. From the parameter deviations, the condition of the machine or the presence of any faults can be concluded.

## 14.2 Next Steps

**Commercial Realization.** As a result of this work, the diagnosis systems have been realized as MATLAB<sup>®</sup> software tools running on a PC, which have been tested on the test bench. For the implementation in a commercial heat pump controller (as a plug-in), the next realization steps will be as follows:

- Select a special machine type.
- Clarify the exact preconditions, such as the number of existing sensors, additional sensors, or available controller hardware (processor, memory, software download options) for the specific machine type. Define the cost limits by a cost-benefit analysis.
- Choose the appropriate diagnosis system (*FuzzyWatch* or *HeatWatch*) and the methods to be applied (*FuzzyWatch*: the *HCM2Neuro* clustering method is recommended)
- Define the realization process. Some suggestions have been made for both diagnosis systems in Section 14.1. Also, the portability of the diagnosis systems has to be checked. In the case of *FuzzyWatch* this means, that the system would be trained for one machine (on the test bench or by simulation), and then it would be validated for another machine of the same type, or even for a machine which is up- or downscaled.
- Implement the software and modify the hardware. The existing MATLAB<sup>®</sup> code has to be converted to realtime code (C code). Therefore, it has to be simplified and an appropriate user interface has to be designed for the controller display. The fast identification methods of the Section below are recommended.

**Reducing the computational effort.** In the case of *FuzzyWatch*, the parameters  $\hat{\theta}$  are identified by a least-squares method. This method needs the calculation of the pseudo-inverse of the regression matrix  $\Phi^T$ , which is filled up with all samples of one data sequence (cf. eq. (19) on page 54). Therefore the data has to be stored during one sequence. This batch algorithm can be replaced by a recursive algorithm, which updates the values of the parameters  $\hat{\theta}$  for each sample by a fast calculation (cf. Appendix A, recursive least-squares identification). Thus, the computational effort as well as the required memory size is significantly reduced.

*HeatWatch* needs the inversion of the regression matrix  $\varphi$ , which contains the mean values of the (quasi) steady-state time interval (cf. eq. (76) on page 145). The matrix inversion can be replaced by a step-wise solution of the linear equation system. The model equations (57)...(62) on page 143 are solved in the following order:

- The refrigerant mass flow  $m_r^*$  is calculated on the basis of the electrical power consumption data,  $P_{HP}$ .
- The heat transfer parameters  $kA_{he,ov}$  and  $kA_c$  are calculated by the eqs. (58) and (59).
- With the parameters above, the heat transfer parameter  $kA_{he,e}$  (eq. (60)) and the mass flow  $m_w^*$  (eq. (62)) are calculated.
- The heat transfer parameter  $kA_e$  is then calculated by eq. (57).
- Finally, the mass flow  $m_s^*$  is calculated by eq. (61)

These equations have to be calculated only once for any data sequence.

**Increasing the sampling time.** In the applications investigated, the sampling time varied between 1 and 10 seconds. For future applications, a larger sampling time would reduce the costs of the online data acquisition module and reduce the required memory size for storing data. The influence of the sampling time onto the diagnosis results was not systematically investigated. But since for the diagnosis methods the signal measurements are not required to be very precise (generally no high frequencies are used for extracting information), a larger sampling time would be possible (maybe using an anti-aliasing filter).

**Extension to the entire heating system.** The diagnosis systems presented in this work are designed for the heat pump as a subsystem. But the same methods can be used for the entire heating system. *FuzzyWatch* can directly be applied to the entire heating system. Therefore, the model is extended by the additional signals (which is very simple) and the additional faults have to be trained. As the experience with the heat pumps described here has shown, it is possible to classify a high number of faults with a low number of sensors. The standard sensors of a heating system only measure the temperatures of the water feed flow and return flow as well as the outdoor temperature. For a detailed fault localization, additional sensors have to be installed at several locations in the heat distribution and emission system (or even the building).

As a first approach for *HeatWatch*, the model is extended with the steady-state equations of the heat load. By using the standard sensors only, the number of parameters will be very low. As a second approach, the parameters of

the linear dynamic model of the heat load can be used (cf. Section 11.2.2), which so far have been identified by an offline algorithm [Shafai 00], [Reiner, Shafai 98]. With an online identification algorithm (ongoing project [Wimmer 01]) some parameters of the heat load can be identified, even if the standard sensors are used only (by using a model of reduced order). For a detailed diagnosis, additional sensors are required as well.

**Other applications.** The presented diagnosis methods can generally be applied to any thermal plants, such as refrigerating machines, air-conditioners, heating or ventilation systems, combined heat-and-power units, or even power plants.

Especially in the case of *FuzzyWatch*, the universal gray-box model approach requires only very small modifications to define the input and output signals. The classification methods are applicable without any modifications. Evidently, the diagnosis system has to be trained with the faults of the new application. In the case of *HeatWatch*, the physical model equations first have to be defined for the new application.

*FuzzyWatch* and *HeatWatch* have been developed for applications that are controlled by an on-off controller. In the heat pump applications investigated, the data sequences are characterized by step responses. If continuous control is used instead, the on-line data processing and evaluation have to be modified, for example by introducing an adaptive parameter identification algorithm (cf. Appendix A, recursive least-squares identification, adaptation).

**Defrosting controller.** The results of either diagnosis system can be used for controlling the activation of the defrosting cycle in an air-to-water heat pump. *FuzzyWatch* has to be trained especially for the fault case ‘evaporator icing’. Specific tests on a test bench are needed to find out which sensors are required for this purpose. As the experience with the other fault cases shows, the number of sensors will be low. The fault case ‘evaporator icing’ can also be added to the existing fault cases, or the existing fault case ‘evaporator fouling’ can just be replaced by the fault case ‘evaporator icing’ (if fouling is low).

Using *HeatWatch*, the heat transfer parameter  $kA_e$  in the evaporator can be used to diagnose icing (cf. Section 9.1). For identifying just the one parameter  $kA_e$ , 3..5 signals are required, which has been evaluated in [Hubacher 00] using the equations of *HeatWatch*. Since this number is too high for one parameter only, it is strongly recommended that all possible parameters are identified for a given set of sensors and that they are used for diagnosis as well.



## 15 References

- [Afjei 01] Th. Afjei, U. Schonhardt, H. P. Eicher, M. Erb, H. R. Gabathuler, H. Mayer, et al.: *Standard hydraulic circuits for small heat pump installations*, interim report, Swiss Federal Office of Energy, 2001, project no. 36228
- [Amann 01] P. Amann: *Identification of fuzzy relational models for fault detection*, Control Engineering Practice, Vol. 9, pp. 555-562, 2001
- [Ayoubi 96] M. Ayoubi: *Fuzzy Systems Design Based on Hybrid Neural Structure and Application to the Fault Dignosis of Technical Processes*, Control Engineering Practice, Vol. 4, No. 1, pp. 35-42, 1996
- [Baehr 92] H. D. Baehr: *Thermodynamik, Eine Einführung in die Grundlagen und ihre technischen Anwendungen*, 8. Auflage, Springer-Verlag, Berlin 1992
- [Birchler 01] B. Birchler, V. Meienhofer: *Messsignale intelligent auswerten – ein Werkzeug für die zustandsabhängige Wartung*, SGA-Bulletin Nr. 32, Schweizerische Gesellschaft für Automatik, 2001
- [Blanke 95] M. Blanke and R. J. Patton, *Industrial Actuator Benchmark for Fault Detection and Isolation*, Control Engineering Practice, Vol. 3, No. 12, pp. 1727-1730, 1995
- [Bott 95] T. R. Bott, *Fouling of Heat Exchangers*, School of Chemical Engineering, University of Birmingham, 1995
- [Chiu 94] S.L. Chiu: *Fuzzy Model Identification Based on Cluster Estimation*, Journal of Intelligent & Fuzzy Systems, Vol. 2, Issue 3, 1994
- [Erb 00] M. Erb, P. Hubacher, H. P. Eicher, M. Farner, *Feldanalyse von WP-Anlagen*, Bundesamt für Energie, 2000
- [Frank 96] P. M. Frank, *Analytical and Qualitative Model-based Fault Diagnosis – A Survey and Some New Results*, European Journal of Control, pp. 6-28, 1996.
- [Frank 00] P. M. Frank, S. X. Ding, B. Köppen-Seliger, *Current Developments in the Theory of FDI*, Preprints of 4th IFAC SAFEPROCESS Symposium, Vol. 1, pp. 16-27, 2000.
- [Fuente 96] M. J. Fuente, P. Vega, M. Zarrop, M. Poch, *Fault Detection in a Real Wastewater Plant Using Parameter Estimation Techniques*, Control Engineering Practice, Vol. 4, No. 8, pp. 1089-1098, 1996.
- [FWS 99] D. Wittwer, S. Berchtold, M. Maspoli: *Wärmepumpen Marktführer Schweiz*, Fördergemeinschaft Wärmepumpen Schweiz (FWS), Auflage März 1999
- [Geering 01] H.P. Geering, *Regelungstechnik*, 5. Auflage, Springer-Verlag, Berlin, 2001

15 References

- [Geering 98] H.P. Geering, *Introduction to Fuzzy Control*, 3<sup>rd</sup> ed., Measurement and Control Laboratory, ETH Zurich, IMRT Press, 1998
- [Gertler 98] J. J. Gertler: *Fault Detection and Diagnosis in Engineering Systems*, Department of Electrical and Computer Engineering, George Mason University (Fairfax, Virginia), Marcel Dekker Inc., 1998
- [Ginsburg 99] S. Ginsburg: *Hierarchische Wärmepumpenregelung mit Fuzzy Control und Robust Control*, Diss. ETH Nr. 13365, Institut für Mess- und Regeltechnik, ETH Zürich, 1999
- [Gruhle 87] W.D. Gruhle: *Regeldynamische Untersuchungen an Kompressionskälteprozessen und -wärmepumpen*, Fortschritt-Berichte VDI, Reihe 19, VDI Verlag, 1987
- [Gubser 99] B. Gubser, L. Wirth, M. Ehrbar: *Dynamischer Wärmepumpentest, Phase 1, Etappe 2, Schlussbericht: Modellbildung*, Bundesamt für Energie, 1999, ENET-9720132/2
- [Guglielmi 95] G. Guglielmi, T. Parisini, G. Rossi, *Fault Diagnosis and Neural Networks: A Power Plant Application*, Control Engineering Practice, Vol. 3, No. 5, pp. 601-620, 1995.
- [Höppner 97] F. Höppner, F. Klawonn, R. Kruse: *Fuzzy-Clusteranalyse: Verfahren für die Bilderkennung, Klassifikation und Datenanalyse*, Vieweg-Verlag, 1997
- [Hubacher 00] B. Hubacher, M. Ehrbar: *Verbesserung des Abtauens bei luftbeaufschlagten Verdampfern, Phase 1: Analyse gängiger Abtauverfahren*, Bundesamt für Energie, 2000, ENET-Nr 9816187
- [IEA 96] J. Hyvärinen (editor): *Real Time Simulation of HVAC Systems for Building Optimization, Fault Detection and Diagnosis*, Technical Papers of IEA Annex 25, International Energy Agency (IEA), Finland, 1996
- [IEA-HP 01/1] Heat Pump Newsletter: *Experience from operating heat pump systems*, International Energy Agency (IEA), Vol. 19, No. 1/2001
- [IEA-HP 01/3] Heat Pump Newsletter: *Mitigating ozone depletion and global warming*, International Energy Agency (IEA), Vol. 19, No. 3/2001
- [Isermann 96] R. Isermann, *Modellgestützte Überwachung und Fehlerdiagnose Technischer Systeme*, atp Automatisierungstechnische Praxis, 5/96, 6/96. Oldenbourg, 1996.
- [Jang 99] R. Jang and N. Gulley, *Fuzzy Logic Toolbox, User's Guide*, The Mathworks Inc., 1999.
- [Klea 95] ICI Klea: *R407C – Physical Property Data Sheet*, ICI Chemicals & Polymers Limited, UK, 1995

- [Koivo 94] H. N. Koivo, *Artificial Neural Networks in Fault Diagnosis and Control*, Control Engineering Practice, Vol. 2, No. 1, pp. 89-101, 1994.
- [Kuijk 96] H. v. Kuijk, *The Application of Field Characteristics to Monitor Heat Pumps for Fault Detection*, Sulzer Energy Consulting, Switzerland, 1996.
- [Kuhlmann 00] C. Kuhlmann, *Instandhaltungsmanagement und Betriebskostenoptimierung für Wärmeaustauscher*, SGA-Bulletin Nr. 28, Schweizerische Gesellschaft für Automatik, 2000.
- [Ljung 87] L. Ljung: *System Identification: Theory for the User*, University of Linköping, Sweden, Prentice-Hall Inc., New Jersey 1987
- [Ljung 98] L. Ljung: *System Identification Toolbox*  
The MathWorks Inc., 1993/1998
- [MathLib 99] MATLAB C++ Math Library©, *User's Guide*, The Mathworks© Inc., 1999
- [Matlab 98] MATLAB, *Using Matlab*, The Mathworks© Inc., 1998
- [Maurer 92] D. E. Maurer, *Expertensystem für die Störungsdiagnose an wärmetechnischen Anlagen von Gebäuden*, Forschungsbericht Nr. 14, Laboratorium für Energiesysteme, ETH Zürich, Juris Verlag, 1992.
- [Maurer 96] D. E. Maurer, *Man Machine Interface for a Failure Diagnosis Expert System*, Technical Papers of IEA Annex 25, pp. 99-115, 1996.
- [Milek 99] J. Milek, O. Hermann, F. Kraus, *Principal Surface Models for Fault Detection, Isolation, and Reconstruction*, Proc. of the 4<sup>th</sup> National Conference in Diagnostics of Industrial Processes, Poland, 1999.
- [Neuensch. 96] P. Neuenschwander, *Wärmetauscher-Überwachung durch Messen von Ein- und Ausgangsgrößen*, Forschungsbericht Nr. 22, Laboratorium für Energiesysteme, ETH Zürich, Juris Verlag, 1996.
- [Patton 00] R. J. Patton, F. J. Uppal, C. J. Lopez-Toribio, *Soft Computing Approaches to Fault Diagnosis for Dynamic Systems: A Survey*, Preprints of 4th IFAC SAFEPROCESS Symposium, Vol. 1, pp. 298-311, 2000.
- [Räber 97] R. Räber, *Spektralmethode zur Fehlerfrüherkennung in wärmetechnischen Anlagen*, Diss. ETH Nr. 12234, Zürich, 1997.
- [Reiner, Shafai 98] G. Reiner, E. Shafai, R. Wimmer, D. Zogg,, H.R. Gabathuler, H. Mayer, H.U. Bruderer: *Kurztestmethode für Wärmepumpenanlagen – Phase 1 bis 3: Messung, Modellierung und Erprobung der Parameteridentifikation*, Bundesamt für Energie, 1998, ENET-Nr. 9657407

15 References

- [Rossi 97] T. M. Rossi and J. E. Braun, *A Statistical, Rule-Based Fault Detection and Diagnostic Method for Vapour Compression Air Conditioners*, HVAC&R Research, Vol. 3, No. 1, pp. 19-37, 1997.
- [Roverso 00] D. Roverso, *Soft computing tools for transient classification*, Information Sciences 127, pp. 137-156, 2000.
- [Shafai 97] E. Shafai: *Adaptive Regelung: Teil I, Parameteridentifikation*, Institut für Mess- und Regeltechnik, ETH Zürich, 1997
- [Shafai 99] E. Shafai, S. Ginsburg, R. Wimmer, D. Zogg, *Validation of a model for a residential heat pump system based on short-period measurements*, Proc. of the 6<sup>th</sup> International Energy Agency Heat Pump Conference, Poster Presentation 42, Berlin, 1999.
- [Shafai 00] E. Shafai, *Kurztestmethode für Wärmepumpen und Gesamtsysteme - 1. Teil*, Tagungsband zur 7. UAW Tagung, Bundesamt für Energie, Burgdorf 2000
- [Söderström 89] T. Söderström, P. Stoica, *System Identification*, Prentice Hall, 1989.
- [Stylianou 96] M. Stylianou, D. Nikanpour, *Performance Monitoring, Fault Detection and Diagnosis of Reciprocating Chiller*, ASHRAE Transactions 1996, Vol. 102, Part 1.
- [Thomson 00] M. Thomson, P. M. Twigg, B. A. Majeed, N. Ruck, *Statistical process control based fault detection of CHP units*, Control Engineering Practice 8, pp. 13-20, 2000.
- [VDI 91] VDI (Herausgeber): *VDI-Wärmeatlas, Verschmutzung von Wärmeübertragerflächen (Abschnitt Oc)*, 6. Auflage, VDI-Verlag 1991
- [Weyer 00] E. Weyer, G. Szederkenyi, K. Hangos, *Grey box fault detection of heat exchangers*, Control Engineering Practice 8, pp. 121-131, 2000.
- [Wimmer 01] R. Wimmer, E. Shafai, H. P. Geering: *Model Predictive Control for Heat Pump Heating Systems*. Proc. of the IASTED International Conference on Control and Applications, Banff, Canada, 2001.
- [Zogg 98] D. Zogg, E. Shafai, *Modellbasierte Fehlerdiagnose an Wärmepumpen*, Bulletin SEV/VSE Nr. 24, Zürich, 1998
- [Zogg 99] D. Zogg, E. Shafai: *Kurztestmethode für Wärmepumpenanlagen – Phase 4: Parameteridentifikation und Fehlerdiagnose für das Teilsystem Wärmepumpe*, Bundesamt für Energie, 1999, ENET No. 981782
- [Zogg 00] D. Zogg, *Kurztestmethode für Wärmepumpen und Gesamtsysteme - 2. Teil*, Tagungsband zur 7. UAW Tagung, Bundesamt für Energie, Burgdorf 2000

- [Zogg 01/1] D. Zogg, E. Shafai, H.P. Geering: *A Fault Diagnosis System for Heat Pumps*, Proc. of the 2001 IEEE Conference on Control Applications (CCA), Mexico City, pp. 70-76
- [Zogg 01/2] D. Zogg, E. Shafai: *A Short-Term Rating Method for Heat Pump Heating Systems – Phase 5: Test of the Fault Diagnosis Systems*, Swiss Federal Office of Energy, 2001, ENET Project No. 37129



# Appendix A: Identification

## Persistent Excitation and Consistency Properties

**Persistent Excitation.** The least-squares solution for the parameters  $\hat{\theta}_{LS}$  in eq. (220) (cf. Section 5.3.1) exists, if  $\Phi\Phi^T$  is invertible, which is the case if the excitation is persistent [Gertler 98] [Ljung 87]. In practice,  $\Phi\Phi^T$  has to be well conditioned, which results in reliable estimates.

$$\hat{\theta}_{LS} = (\Phi\Phi^T)^{-1}\Phi \cdot Y \quad (220)$$

Some reasons for rank defects, in the case of linearly dependent columns in  $\Phi^T$  and thus in  $\Phi\Phi^T$ , are listed below.

- Repetitions in the sequence would result in linearly dependent columns of  $\Phi^T$ . The probability of repetitions in the measured signals is low, since only one data sequence (off-on) is evaluated for identifying one parameter set.
- Linear dependence among multiple inputs would also result in a rank defect. The use of real data from the nonlinear process reduces this problem. Using the data from a simulation model, the correlations generally are higher (cf. Section 6.2). But with the nonlinear simulation model used here, the signal correlations are sufficiently low.
- Over-parametrization of the model would lead to linear dependence. This is avoided by only considering low-order models, which are first-order ARX models or even static models (cf. Section 5.2.4).
- Data collection under feedback control can also lead to rank defect. The heat pump is internally controlled by the thermostatic expansion valve, which is a P-controller (with saturations). This controller directly influences the mass flow, which is not measured. The measured temperature signals are only indirectly influenced by the mass flow in a nonlinear way. The external power control is realized by an on/off controller, which has no influence during one data sequence.

Nevertheless, some badly conditioned matrices may occur. Therefore, only the best model approaches are selected during the training phase, by comparing the model errors (cf. Section 5.3.3). If the error sizes are in the same region for two approaches, the model with a lower order is preferred (in order to reduce the probability of over-parametrization). If the error sizes are large, then no model is selected for the corresponding output signal.

**Consistency.** The least-squares solution in eq. (220) for  $n_k=K$  measurements can also be expressed by eq. (222).

$$y(t) = \varphi^T(t) \cdot \theta + \varepsilon(t) \quad (221)$$

$$\hat{\theta}_{LS}(t, K) = \left[ \sum_{k=0}^{K-1} \varphi(t-k) \varphi^T(t-k) \right]^{-1} \sum_{k=0}^{K-1} \varphi(t-k) y(t-k) \quad (222)$$

Then the consistency error for an infinite set of observations ( $K \rightarrow \infty$ ) is given by eq. (223) with the true parameter vector  $\theta_0$  and the equation error  $\varepsilon(t)$  (eq. (221)) [Gertler 98], [Ljung 87].

$$\hat{\theta}_{LS}(t, \infty) - \theta_0 = \underbrace{\left[ \lim_{K \rightarrow \infty} \frac{1}{K} \sum_{k=0}^{K-1} \varphi(t-k) \varphi^T(t-k) \right]^{-1}}_{\neq 0 \Rightarrow \text{persistent}} \underbrace{\left[ \lim_{K \rightarrow \infty} \frac{1}{K} \sum_{k=0}^{K-1} \varphi(t-k) \varepsilon(t-k) \right]}_{=0 \Rightarrow \text{consistent}} \quad (223)$$

For an unbiased estimation the consistency error has to be zero, which is the case if the equation error  $\varepsilon(t)$  has zero mean, is uncorrelated with the input sequence, and is itself an uncorrelated (white) sequence. Under finite datasets the estimation is generally biased. Since for the measured signals of the actual applications, the equation error  $\varepsilon(t)$  is not white noise and the length of the datasets is finite, the identified parameters are biased. But for the fault diagnosis systems the bias is not a principal problem, since the real parameter values  $\theta_0$  are not used. The bias might change from one data sequence to the other. In the fault sensitivity analysis in Section 5.3.4, the parameter charts allow to analyse the sensitivities and the variances even with the bias. Nevertheless, the sensitivities are sufficiently high.

## Recursive Least-Squares Identification

The recursive least-squares (RLS) parameter identification method could optionally be used for online implementation in embedded systems, where the memory and the computational power is restricted. In [Gertler 98], [Ljung 87], and [Shafai 97], the recursive identification method below is presented. Here it is derived from the ‘batch’ algorithm, which is used in Section 5.3.1.

The linear regression form is given by eq. (224).

$$y(t) = \varphi^T(t) \cdot \theta + \varepsilon^T(t) \quad (224)$$

The ‘batch’ algorithm for identifying the parameters from a data sequence with  $K$  measurements is then defined by eqs. (225) and (226).

$$Y = \Phi^T \cdot \theta + E^T \quad (225)$$

$$\hat{\theta} = (\Phi\Phi^T)^{-1} \Phi \cdot Y \quad (226)$$

The batch algorithm may require significant memory, since all data samples of one sequence have to be stored in  $\Phi^T$  and  $Y$ .

Eq. (226) can be expressed by eq. (227) (with the same number of samples  $K$ ), which is rewritten in eq. (230) using the square matrix  $Q(t,K)$  and the vector  $p(t,K)$ .

$$\hat{\theta}(t, K) = \left[ \sum_{k=0}^{K-1} \varphi(t-k)\varphi^T(t-k) \right]^{-1} \sum_{k=0}^{K-1} \varphi(t-k)y(t-k) \quad (227)$$

$$Q(t, K) = \sum_{k=0}^{K-1} \varphi(t-k)\varphi^T(t-k) \quad (228)$$

$$p(t, K) = \sum_{k=0}^{K-1} \varphi(t-k)y(t-k) \quad (229)$$

$$\hat{\theta}(t, K) = Q^{-1}(t, K) \cdot p(t, K) \quad (230)$$

With a recursive data collection,  $Q(t,K)$  and  $p(t,K)$  are updated for each data sample (eqs.(231) and (232)).

$$Q(t+1, K+1) = Q(t, K) + \varphi(t+1)\varphi^T(t+1) \quad (231)$$

$$p(t+1, K+1) = p(t, K) + \varphi(t+1)y(t+1) \quad (232)$$

Thus, it becomes unnecessary to form the large  $\Phi^T$  and  $Y$ , which helps to save memory. These equations can be used to collect the data during one sequence. At the end of the sequence, the parameters are computed once by eq. (230). But in eq. (230), the inversion of the matrix  $Q$  is still necessary, which requires some computational effort.

In the next step, a recursive matrix inversion is introduced (eq. (233)), which, by using some standard formula, is the inversion of eq. (231). Thus, the computational effort is further reduced. For the diagnosis system of this work (*FuzzyWatch*), the use of eq. (233) and eq. (232) for each sample, together with the direct application of eq. (230) at the end of the data sequence, are

recommended (recursive data collection with ‘batch’ parameter computation).

$$Q^{-1}(t+1, K+1) = Q^{-1}(t, K) - \frac{Q^{-1}(t, K)\varphi(t+1)\varphi^T(t+1)Q^{-1}(t, K)}{1 + \varphi^T(t+1)Q^{-1}(t, K)\varphi(t+1)} \quad (233)$$

Some initial values for  $Q^{-1}(0,0)$  and  $p(0,0)$  have to be defined. For example, they could be defined by the final values of the last data sequence (or an off-line batch run for the first value).

**Optional:**

Using some algebra, the recursive parameter update in eq. (234) is calculated from the equations above. The parameter estimates are corrected by the prediction error (eq. (235)).

$$\hat{\theta}(t+1, K+1) = \hat{\theta}(t, K) + Q^{-1}(t+1, K+1)\varphi(t+1)e(t+1) \quad (234)$$

$$e(t+1) = y(t+1) - \varphi^T(t+1)\hat{\theta}(t, K) \quad (235)$$

For computation, eq. (233) is also needed to calculate the updated inverses of  $Q$ . Thus it is not more effective than the direct application of eq. (230) with the recursive inverse computation of eq. (233)

**Adaptation:**

For an adaptation of the parameters, an exponential forgetting with “forgetting factor”  $\lambda$  is introduced (eqs. (236) and (237)).

$$Q(t+1, \lambda) = \lambda \cdot Q(t, \lambda) + \varphi(t+1)\varphi^T(t+1) \quad (236)$$

$$p(t+1, \lambda) = \lambda \cdot p(t, \lambda) + \varphi(t+1)y(t+1) \quad (237)$$

During the batch data sequences of this work, an adaptation makes no sense. But if the machine is controlled by a continuous controller instead of an on-off controller, then an adaptive parameter identification would be needed.

## Appendix B: From Residuals to Parameters

**Residuals.** Various approaches for model-based residual generation are described in [Gertler 98]. Here the symptoms (cf. Section 3.2) are represented by residuals.

The residuals  $r(t)$  are generated from the measured inputs  $u(t)$  and outputs  $y(t)$  of the process (parity relations). The computational form of the residual generator is given by eq. (238), where  $M(\phi, \theta_0)$  is the nominal model of the process for the fault-free case and  $W(\phi)$  is a “weighting” filter.

$$r(t) = W(\phi)[y(t) - M(\phi, \theta_0)u(t)] \quad (238)$$

Thus, the process output is compared with the output of the nominal model, and a fault is represented by an increase in the residuals. Therefore, the nominal model for the fault-free case has to be known, for example by identification of the nominal parameters  $\theta_0$  (training of the nominal case).

The internal form of the residual generator is given by eq. (239), with the sensitivities  $S_F(\phi)$  to *additive* faults  $f(t)$  and the sensitivities  $S_D(\phi)$  to the disturbances or noise  $d(t)$ .

$$r(t) = W(\phi)[S_F(\phi)f(t) + S_D(\phi)d(t)] \quad (239)$$

For the design, the specifications  $Z_F(\phi)$  and  $Z_D(\phi)$  are defined and eq. (240) has to be solved for  $W(\phi)$ . With  $Z_F(\phi)$  as a diagonal matrix, the residuals are decoupled, and with  $Z_D(\phi)$  as a zero matrix, the disturbances are rejected.

$$W(\phi)[S_F(\phi)f(t) \quad S_D(\phi)d(t)] = [Z_F(\phi) \quad Z_D(\phi)] \quad (240)$$

The fault sensitivities  $S_F(\phi)$  and the sensitivities  $S_D(\phi)$  to the disturbances or noise have to be known from physical considerations or have to be estimated from experiments (training of the fault cases and disturbances).

Since in the application of this work and many other applications, most of the faults are *multiplicative*, the internal form of the residual generator has to be modified (eq. (241)). The parameter changes  $\Delta\theta_F$  are due to the faults and the parameter changes  $\Delta\theta_D$  are due to the disturbances or noise (influencing the same parameters). The model  $M(\phi, \theta)$  is linearized in the parameters  $\theta$  with the linearization matrix  $N(t)$  (eq. (242)).

$$r(t) = W(\phi)[N(t)\Delta\theta_F + N(t)\Delta\theta_D] \quad (241)$$

$$N(t) = \left[ \begin{array}{ccc} \frac{\partial M(\phi, \theta)}{\partial \theta_1} u(t) & \dots & \frac{\partial M(\phi, \theta)}{\partial \theta_{n_\theta}} u(t) \end{array} \right] \quad (242)$$

**Parameters.** A much simpler approach for handling model parameters is the linear regression form (eq. (243)) with the one-step ahead prediction output  $y^*(t)$ , the regression vector  $\varphi^T(t)$ , containing the old outputs and the inputs, and the parameters  $\theta$ .

$$y^*(t) = \varphi^T(t) \cdot \theta \quad (243)$$

In a similar way to the one described above, some primary residuals  $o^*(t)$  can be defined by eq. (244). The corresponding residuals are calculated by  $r^*(t) = W(\phi) o^*(t)$ . Thus, the residuals are expressed by the parameter deviations  $\Delta\theta$ , which is the sum of the parts  $\Delta\theta_F$  and  $\Delta\theta_D$ .

$$\begin{aligned} o^*(t) &= \varphi^T(t) \cdot \theta - \varphi^T(t) \cdot \theta_0 = \varphi^T(t) \cdot \Delta\theta \\ &= \varphi^T(t) \cdot \Delta\theta_F + \varphi^T(t) \cdot \Delta\theta_D \end{aligned} \quad (244)$$

For a data sequence with  $K$  measurements, eq. (244) can be solved for the parameter deviations  $\Delta\theta$  by the least-squares method. For the diagnosis systems of this work, the least-squares method is directly applied to eq. (243) for identifying the parameters  $\theta$ , from which the parameter deviations  $\Delta\theta = \theta - \theta_0$  are calculated afterwards.

**Parameter Sensitivities.** So far, the multiplicative faults and disturbances have been expressed by the parameter changes  $\Delta\theta_F$  and  $\Delta\theta_D$ . In reality, the parameters  $\theta$  are a nonlinear function  $h$  of the faults  $f$  and the disturbances  $d$  (eq. (245)). The parameter sensitivities  $S_F^\theta$  to the faults and the parameter sensitivities  $S_D^\theta$  to the disturbances are expressed by linearization (eq. (246)).

$$\theta = h(f, d) \quad (245)$$

$$\Delta\theta_F \approx \frac{\partial h}{\partial f} \Delta f = S_F^\theta \cdot \Delta f \quad \Delta\theta_D \approx \frac{\partial h}{\partial d} \Delta d = S_D^\theta \cdot \Delta d \quad (246)$$

Comparing this result to the sensitivity analysis of Section 5.3.4, the mean parameters  $\Delta\bar{\theta}_i$  are defined by the faults, and the uncertainties  $\delta\theta$  are defined by the disturbances or noise (cf. eq. (247)). These properties are visualized in the parameter charts.

$$\Delta\bar{\theta}_i \approx \Delta\theta_F \quad \delta\theta \approx \Delta\theta_D \quad (247)$$

**Clustering.** In the case of the residual approach, the parameters  $\theta_0$  of the nominal model have to be identified, which corresponds to finding the location of the nominal cluster in the parameter space during the training phase (cf. Section 5.4). In the case of the residual approach, the fault sensitivities  $S_F(\phi)$  have to be estimated from experiments (if they are not known), which

corresponds to finding the locations of the fault clusters in the parameter space. Estimating the sensitivities  $S_D(\phi)$  to the disturbances or noise is replaced by estimating the extents of the fault clusters (statistical properties like standard deviations). The decoupling and disturbance rejection problem is solved by the clustering algorithms.

**Thresholds.** For the residuals, thresholds have to be defined which cause an alarm. Defining the threshold for each residual is a sophisticated procedure in order to minimize false alarms and missed detections. Sometimes, constant thresholds are not sufficient, which leads to adaptive thresholds. In the diagnosis system of Chapter 5 (*FuzzyWatch*), the threshold problem is less severe. Here only the nominal weighting parameter or the minimal fault grade parameter is used to minimize false alarms (cf. Section False Alarm Tuning), else no thresholds are needed. In the diagnosis system of Chapter 8 (*HeatWatch*), no thresholds are defined, since it is restricted to displaying the deviations of the current parameters without causing any alarms. If alarming is required, thresholds would have to be defined for each parameter.

**Comparison.** Although the approaches used in this work seem to be quite different from the residual approach, they result in similar considerations. For the applications of this work, the parameter identification approach has the following advantages:

- Only one parameter set is identified for one data sequence instead of a sequence of residuals, which results in a high data reduction.
- In the heat pump and many other applications most faults are multiplicative, which can be easily handled with parameter identification methods.
- In the case of physical parameters (*HeatWatch*), the relation between the faults and the parameters is known. Only one parameter is or a few parameters are significantly influenced by one fault and no decoupling problem has to be solved.
- The parameter sensitivities can be easily estimated by using the clustering algorithms (*FuzzyWatch*).

A disadvantage of the parameter identification method for on-line use is its computational effort. With the methods of Appendix A (recursive least-squares identification) the effort is significantly reduced.



## Appendix C: Data Sheets

### SATAG Natura AW110

The Natura air-to-water heat pump from SATAG<sup>®</sup> Thermotechnik AG is installed in the residential building at Barzheim. The heat pump contains an additional hot gas defrosting cycle and an internal heat exchanger.

Leistungsdaten in kW	 <b>AW 110.1 Hi</b>		
Betriebspunkt	A2 / W35	A-5 / W45	A-8 / W45
Heizleistung	9.3 <sup>(1)</sup>	8.0	7.5
Kälteleistung (Umweltwärme)	6.5	4.6	4.1
el. Aufnahmeleistung	2.8 <sup>(2)</sup>	3.4	3.4
<b>COP</b> (Leistungszahl)	3.31	2.33	2.18

Elektrische Daten	Einheit	Wert
Nennspannung	V	3 x 400
Betriebsstrom max.	A	7.9
Anlaufstrom	A	45 (30 <sup>(3)</sup> )
Blockierstrom LRA	A	51
Absicherung in der Zuleitung	A	16 träge

Wärmeabgabe	Einheit	Wert
Heizwasserdurchsatz	l/h	950
Heizwasseranschluss	Zoll / <i>pouces</i>	1
Druckverlust Kondensator	mbar	30
Vorlauftemperatur max.	°C	45 (A-15)
	°C	55 (A-5)

Wärmegewinnung	Einheit	Wert
Ventilatorleistung	W	220
Luftmenge	m <sup>3</sup> /h	3500
max. zul. Druckverlust Zuluft- + Abluftkanal	Pa	30
Lufttemperatur min.	°C	-15
Abtauleistung	W	ca. 3300
Anteil Abtauzeit von Laufzeit	%	7 - 17

Wärmepumpe	Einheit	Wert
Abmessungen (Höhe x Breite x Tiefe)	mm	1470 / 1200 / 760
Geräusch (Freifeld 5m)	dB	45
Kompressoren vollhermetisch	Stk. / pcs.	1
Arbeitsmittelfüllung <b>R407C</b>	kg	4.2
Gewicht	kg	250

<sup>(1)</sup> Nenn-Heizleistung nach EN 255

<sup>(2)</sup> el. Nennleistung  $P_{NT}$

<sup>(3)</sup> mit Anlaufstrombegrenzer

## SATAG Natura BW108

The Natura brine-to-water heat pump from SATAG<sup>®</sup> Thermotechnik AG is used in the test bench at the Measurement and Control Laboratory (IMRT). The heat pump contains an internal heat exchanger.

Leistungsdaten in kW			
Betriebspunkt	B0 / W35	B2 / W45	B2 / W55
Heizleistung	8.3 <sup>(1)</sup>	8.5	8.1
Kälteleistung (Umweltwärme)	6.50	6.25	5.30
el. Aufnahmeleistung	1.80 <sup>(2)</sup>	2.25	2.75
<b>COP</b> (Leistungszahl)	4.61	3.77	2.95

Elektrische Daten	Einheit	Wert
Nennspannung	V	3 x 400
Betriebsstrom max.	A	6.6
Anlaufstrom	A	39 (20 <sup>(3)</sup> )
Blockierstrom LRA	A	43.5
Absicherung in der Zuleitung	A	16 träge

Wärmeabgabe	Einheit	Wert
Heizwasserdurchsatz	l/h	700
Heizwasseranschluss	Zoll / pouces	1
Druckverlust Kondensator	mbar	40
Vorlauftemperatur max.	°C	55

Wärmege winnung	Einheit	Wert
Sole-Durchsatz	l/h	2100
Sole-Anschluss	Zoll / <i>pouces</i>	1
Druckverlust Verdampfer	mbar	90
max. Sole-Eintrittstemperatur (bei höherer Temp. Mischer vorschalten)	°C	25
min. Sole-Eintrittstemperatur	°C	-5
Frostschutzkonzentration (Aethylenglykol)		
- für Erdwärmesonden	%	23
- für Flachregister	%	33

Wärmepumpe	Einheit	Wert
Abmessungen (Höhe x Breite x Tiefe)	mm	970 / 610 / 610
Kompressoren vollhermetisch	Stk. / <i>pces.</i>	1
Arbeitsmittelfüllung <b>R407c</b>	kg	2.2
Gewicht	kg	120

<sup>(1)</sup> Nenn-Heizleistung nach EN 255

<sup>(2)</sup> el. Nennleistung  $P_{NT}$

<sup>(3)</sup> mit Anlaufstrombegrenzer

## ZHW Laboratory Heat Pump

The laboratory brine-to-water heat pump is integrated in the test bench at the University of Applied Sciences at Winterthur (ZHW). The heat pump contains an additional subcooler.

Operating point	Unit	Value
evaporation temperature	°C	0
superheating temperature difference	K	10
condensation temperature	°C	50
power supply frequency	Hz	50
compressor speed	r.p.m.	1450
refrigerant		R134a
brine mixture: ethylen glycol / water	volumetric %	34

Appendix C: Data Sheets

<b>Evaporator</b>	<b>Unit</b>	<b>Value</b>
evaporator type		plates
heat exchanger area	m <sup>2</sup>	2.14
brine input temperature	°C	+8
brine outlet temperature	°C	+3.5
brine mass flow	kg/h	2045
cooling power	kW	9.6

<b>Compressor</b>	<b>Unit</b>	<b>Value</b>
compressor type		piston
number of cylinders		2
refrigerant accumulator volume	l	15
volumetric flow (theoretic value)	l/h	19'600
oil well heating power	W	70
speed range	r.p.m.	750..1750
frequency range	Hz	25..60
power consumption	kW	3

<b>Condenser</b>	<b>Unit</b>	<b>Value</b>
condenser type		plates
number of plates		20
heat exchanger area	m <sup>2</sup>	0.65
water input temperature	°C	37.5
water outlet temperature	°C	46.5
water mass flow	kg/h	1040
heating power	kW	11

<b>Subcooler</b>	<b>Unit</b>	<b>Value</b>
heat exchanger type		plates
number of plates		10
heat exchanger area	m <sup>2</sup>	0.29
water input temperature	°C	30
water outlet temperature	°C	40
water mass flow	kg/h	137
heating power	kW	1.6

## Simulation Model Parameters

The simulation model is adapted to the SATAG<sup>®</sup> Natura AW110 air-to-water heat pump, which is installed in the residential building at Barzheim. The parameters are either given by the manufacturer, or they are estimated (estimated parameters marked with ‘\*’ in the tables below). The parameters at the interface of the heat pump, such as the air flow or the water flow, are measured at the particular building at Barzheim (marked with ‘+’). Certain parameters have been adjusted in order to fit the measured data (marked in **bold** in the tables below). In order to adjust the heating power output, the power consumption and the other heat flows, a power scaling factor (*sc*) has been introduced.

The three parameter sets for each operation mode (on, off, defrosting) mainly differ in the adjusted parameters (in **bold**). For these parameters, only the values for the normal operation mode are listed below. However, some additional parameters have been defined for the other modes (off, defrosting), which are marked in *italics*. Note that, for practical reasons, the parameters of each module (evaporator, internal heat exchanger, ...) are combined in individual tables below, whereas in Chapter 11, the equations of each process (evaporation, superheating, ...) are combined in individual Sections.

Heat Pump Power Scaling	Symbol	Unit	Value
scaling factor	<i>sc</i>	-	1.6

Evaporator	Symbol	Unit	Value
evaporator type			bundles of pipes
number of parallel pipes			11
cross-sectional area (sum over all pipes)	$A_e$	mm <sup>2</sup>	552.6 · <i>sc</i>
total length of one pipe	$L_{e,tot}$	m	15
*mean liquid cross-sectional area ratio	$\beta_e$		0.5
*mass transfer rate	$k_{tr,e}$	kg/(s·bar)	1
* <b>evaporation heat transfer parameter</b>	$kb_{e,e}$	W/(m·K)	0.12 · <i>sc</i>
* <b>superheating heat transfer parameter</b>	$kb_{e,ov}$	W/(m·K)	0.01 · <i>sc</i>
* <b>inertia of liquid phase</b>	$\tau_{e,l}$	J/bar	300
+air flow	$V_a^*$	m <sup>3</sup> /s	1.3
	$m_a^*$	kg/s	1.7
* <b>inertia of air side</b>	$\tau_a$	J/K	50

Internal Heat Exchanger	Symbol	Unit	Value
heat exchanger type			coaxial
*total length	$L_{he,tot}$	m	1
*evaporation heat transfer parameter	$kb_{he,e}$	W/(m·K)	0.007·sc
*superheating heat transfer parameter	$kb_{he,ov}$	W/(m·K)	0.04·sc
*inertia of superheated gas	$\tau_{ov}$	J/K	0.1
*inertia of subcooled liquid	$\tau_{sc}$	J/K	1

Compressor	Symbol	Unit	Value
compressor type			scroll
*characteristics (pressure-to-mass-flow)	$a_{cmp}$		0.06·sc
	$b_{cmp}$		-0.4
	$c_{cmp}$		0.15
	$d_{cmp}$		0
*mass flow supply efficiency	$\lambda_{cmp}$	-	0.9
*mode off: backflow valve constant	$k_{cmp}$	kg/s·bar <sup>-0.5</sup>	1.6·10 <sup>-4</sup>
*electro-mechanical efficiency	$\eta_{cmp}$	-	0.8
*thermal inertia of compressed gas	$\tau_{cmp}$	J/K	20

Condenser	Symbol	Unit	Value
condenser type			plate
number of plates			40
*cross-sectional area (sum between all plates, scaled with 2)	$A_c$	m <sup>2</sup>	0.002·2·sc
total height (scaled with 2)	$L_{c,tot}$	m	0.5·2 = 1
*mass transfer rate	$k_{tr,c}$	kg/(s·bar)	1
*condensation heat transfer parameter	$kb_{c,c}$	W/(m·K)	20·sc
*hot gas cooling heat transfer param.	$kb_{c,g}$	W/(m·K)	0.01·sc
*inertia of liquid phase	$\tau_{c,l}$	J/bar	0.5
*inertia of hot gas	$\tau_{c,g}$	J/K	0.1
+ water flow	$V_w^*$	l/h	1'400
	$m_w^*$	kg/s	0.4
*inertia of water side	$\tau_w$	J/K	50

Expansion Valve / Hot-Gas valve	Symbol	Unit	Value
valve type (superheating controller)			thermostatic
*superheating setpoint			10
<b>*controller settings</b>	$a_{exp}$		0.1
	$b_{exp}$		0.36
<b>*characteristics, valve constants</b>	$k_{exp,l}$	kg/s·bar <sup>-0.5</sup>	0.025·sc
	$k_{exp,g}$	kg/s·bar <sup>-0.5</sup>	0.03·sc
*mode defrosting: hot-gas valve constant	$k_{hg}$	kg/s·bar <sup>-0.5</sup>	0.016·sc
*mode defrosting: condensation ratio	$k_{hg,c}$	-	1

Icing of the Evaporator	Symbol	Unit	Value
° = values given by [Ginsburg 99]			
<b>*scaling factor for icing effect</b>	$sc_{ice}$		4
*ice building parameter	$k_{ice,build}$	-	0.02·sc <sub>ice</sub>
*self-defrosting parameter	$k_{ice,selfdefr}$	-	10 <sup>-5</sup> ·sc <sub>ice</sub>
*enforced defrosting parameter	$k_{ice,selfdefr}$	-	0.2·sc <sub>ice</sub>
°temperature level for ice building	$T_{ice,build}$	°C	1
°temperature level for self-defrosting	$T_{ice,selfdefr}$	°C	0
*heat transfer reduction parameter	$a_{ice}$	1/kg	1
°pressure drop offset (= value without ice)	$\Delta p_{a,0}$	mbar	0.17
<b>°pressure drop gain</b>	$k_{\Delta p}$	mbar/kg	5 / sc <sub>ice</sub>

The physical property data of the refrigerant R407C is defined by the polynomials in Section 11.1.1. The parameters below are given by the refrigerant manufacturer [Klea 95].

Refrigerant Data R407C	Symbols	Values
critical point, eq. (186)	$T_c, p_c, \rho_c$	359 K, 46.5 bar, 490 kg/m <sup>3</sup>
equation of state, eq. (187)	$X, K$	3.65, 5.48
equation of state, eq. (187)	$A_l, B_l, C_l$	-11.8, 6.96, -12.9
bubble-point temp., eq. (188)	$A_b, B_b, C_b, D_b$	229, 20.9, 1.85, 0.378
mid-point temp., eq. (189)	$A_m, B_m, C_m, D_m$	232, 20.6, 2.05, 0.308
dew-point temp., eq. (190)	$A_d, B_d, C_d, D_d$	236, 20.3, 2.18, 0.250
sat. liquid enthalpy, eq. (191)	$A', B', C', D', E'$	132, 1225, -4212, 5045, -2528
latent heat of vap., eq. (194)	$A_r, B_r, C_r, D_r, E_r$	219, -1683, 6325, -8231, 3884
gas heat capacity, eq. (198)	$A_g, B_g, C_g, D_g$	-0.084, 0.0032, -1.94·10 <sup>-6</sup> , 36.4
liquid density, eq. (200)	$A'_\rho, B'_\rho, C'_\rho, D'_\rho, E'_\rho$	-650, 7862, -15312, 15705, -5734
saturated vapour density, eq. (201)	$A''_\rho, B''_\rho, C''_\rho, D''_\rho, E''_\rho$	-650, 7862, -15312, 15705, -5734

The building model is validated for the residential building at Barzheim [Reiner, Shafai 98].

<b>Building Model</b>	<b>Symbol</b>	<b>Unit</b>	<b>Value</b>
all values given by [Reiner, Shafai 98]			
heat transfer from heating water to floor	$kA_{w,f}$	W/K	978.3
heat transfer from floor to room	$kA_{f,r}$	W/K	666.3
heat transfer from room to environment	$kA_{r,amb}$	W/K	104.6
total volume of heating water	$V_w$	m <sup>3</sup>	0.19
inertia of the floor	$mC_f$	J/K	$1.03 \cdot 10^7$
time constant of the building	$\tau_r$	h	95.8

## **Curriculum Vitae**

### **Personal Data**

Name: David Zogg  
Date of Birth: January 15, 1973  
Birth place: Winterthur, Switzerland

### **Education**

1980-1986 Primary and secondary schools at Oberburg BE  
1986-1993 Gymnasium at Burgdorf BE  
1993 Matura certificate, type C  
1993-1995 Basic studies of mechanical engineering, Swiss Federal Institute of Technology (ETH) Zürich  
1996 Practice in industry at Sulzer Innotec, thermodynamics of fuel cell stacks  
1996-1997 Diploma studies of mechanical engineering, Swiss Federal Institute of Technology (ETH) Zürich  
Majors in measurement and control theories and turbo machinery  
1997-1998 Diploma thesis in industry at Asea Brown Boveri (ABB), controller design for gas turbines  
1998 Diploma as Dipl. Masch.-Ing. ETH  
1998-2002 Doctoral student, research and teaching assistant at the Measurement and Control Laboratory (IMRT), ETH Zurich, Switzerland