
Pontificia Universidad Católica del Perú
Escuela de Posgrado

Tesis de Maestría

Model-Based Fault Diagnosis
via Structural Analysis
of a Reverse Osmosis Plant

Para optar el grado académico de
Magíster en Ingeniería de Control y Automatización

Presentado por:
Johannes Göpfert

Profesor Responsable en TU Ilmenau: Prof. Dr.-Ing. Johann Reger

Profesor Responsable en PUCP: Prof. Dr. Carlos Gustavo Pérez Zúñiga

Febrero, 2021

Abstract

Water desalination is one approach to force water scarcity. One of the processes used for desalination is reverse osmosis. Like other systems, a reverse osmosis plant is susceptible to faults. A fault can lead to a loss of efficiency, or if the fault is severe to a total breakdown. Appropriate measures can minimize the impact of faults, but this requires in time fault detection.

The following thesis shows a proposal for an online fault diagnosis system of a reverse osmosis plant. For the model-based approach, a mathematical model of a reverse osmosis plant has been developed. The model contains a new approach for modeling the interaction between the high-pressure pump, the brine valve, and the membrane module. Furthermore, six faults considered for fault diagnosis have been modeled. Two of the faults are plant faults: The leakage of the feed stream and membrane fouling. The other four faults are sensor or actuator malfunctions.

The fault diagnosis system is developed via structural analysis, a graph-based approach to determine a mathematical model's overdetermined systems of equations.

With the structural analysis, 73 fault-driven minimal structurally overdetermined (FMSO) sets have been determined. The results show that all six faults are detectable. However, two faults are not isolable. Five of the FMSO sets have been chosen to deduce the residuals used for online fault detection and isolation. The simulations demonstrate that the calculated residuals are appropriate to detect and isolate the faults. If one assumes that only the considered faults occur, it is possible to determine some faults' magnitude.

Model-Based Fault Diagnosis via Structural Analysis of a Reverse Osmosis Plant

Master Thesis

Submitted by: Johannes Göpfert

Date and place of birth: 06.03.1990 in Karlstadt, Germany

Student number: 20197174 (PUCP) /
54309 (TU Ilmenau)

Course of studies: Ingeniería de Control y Automatización (PUCP) /
Technische Kybernetik und Systemtheorie (TU Ilmenau)

Department: Escuela de Posgrado (PUCP) /
Regelungstechnik (TU Ilmenau)

Supervisors: Prof. Dr. Carlos Gustavo Pérez Zúñiga (PUCP) /
Prof. Dr.-Ing. Johann Reger (TU Ilmenau)

Date and place: February 25, 2021, Zellingen

Declaration

I, Johannes Göpfert, hereby declare that the work is entirely my own and was produced with no assistance from third parties.

Formulations and ideas taken from other sources are cited as such.

This work has not been published and submitted, either in part or whole, for obtaining a degree at this or any other university.

Zellingen, February 25, 2021

Johannes Göpfert

Contents

Abstract	i
Declaration	ii
List of Figures	iii
List of Symbols	iv
List of Abbreviations	v
1 Introduction	1
2 State of the art	3
2.1 Water desalination	4
2.1.1 Multi - Stage Flash distillation	6
2.1.2 Reverse osmosis	7
2.2 Reverse osmosis plant	9
2.3 Fault diagnosis	13
2.3.1 System behavior	13
2.3.2 Consistency based fault diagnosis	14
2.3.3 Process models	15
2.3.4 Fault detection methods	17
2.4 Recent works	21
2.4.1 Existing mathematical models of reverse osmosis plants	21
2.4.2 Existing fault diagnosis systems for reverse osmosis plants	24
3 Mathematical modeling of the reverse osmosis plant	28
3.1 The model structure	29

3.2	The membrane module	30
3.2.1	Mathematical model of the membrane	32
3.3	The high pressure pump module	34
3.4	Modeling the hydraulic circuit	36
3.5	Measured and known variables	40
3.6	Faults	43
3.6.1	Multiplicative and additive faults	43
3.6.2	Faults of the reverse osmosis plant	44
3.7	Model evaluation and simulation	46
4	Fault diagnosis via structural analysis	53
4.1	Structural model	54
4.2	Matching	58
4.3	Dulmage-Mendelsohn decomposition	61
4.4	FMSO algorithm	69
4.4.1	An algorithm for the Dulmage-Mendelsohn decomposition	69
4.4.2	An algorithm for finding FMSOs	72
4.5	Structural analysis of the RO system	75
4.5.1	FMSO selection	78
5	Implementation proposal	84
5.1	Deducing the ARRs	85
5.1.1	ARR of FMSO 2	85
5.1.2	ARR of FMSO 21	87
5.1.3	ARR of FMSO 30	88
5.2	Simulation of the fault diagnosis system	91
6	Conclusion	98
	Bibliography	100
A	All found FMSOs	106

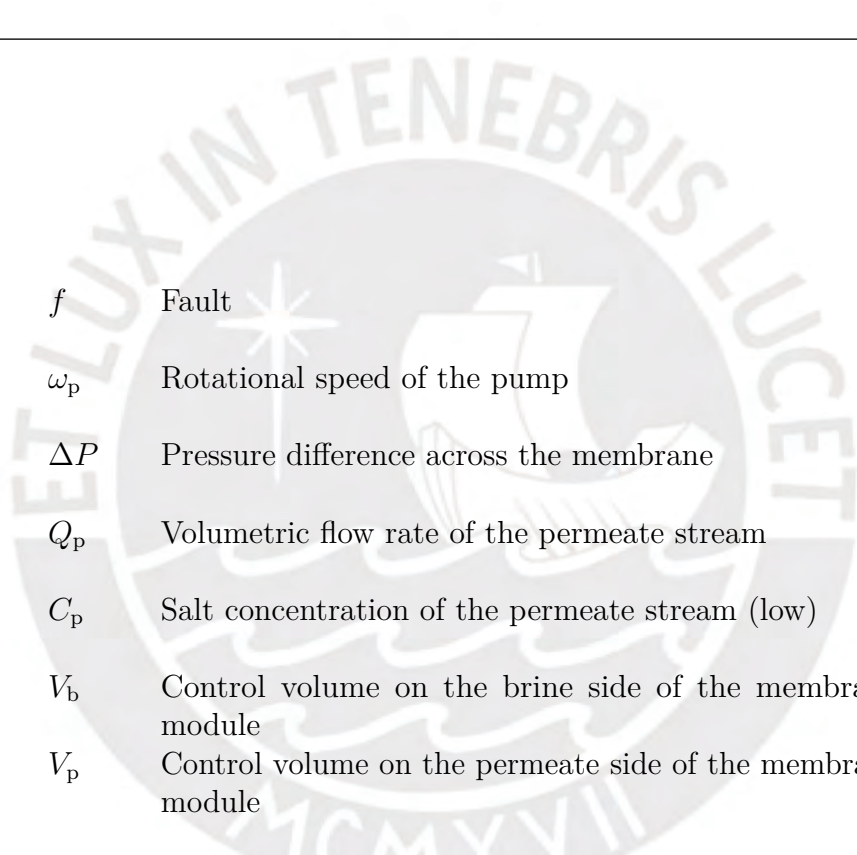
List of Figures

2.1	Classification of water desalination processes [SR14]	5
2.2	Schematic of MSF process [Naj16]	6
2.3	Process of osmosis [EE02]	8
2.4	Process of reverse osmosis [EE02]	8
2.5	Reverse osmosis pilot plant [RSP ⁺ 19]	9
2.6	Hollow fiber (a) and spiral wound (b) membrane module [Kuc15]	11
2.7	Graphical illustration of the system behavior [BKL ⁺ 16]	13
2.8	Change of behavior for a faulty system [BKL ⁺ 16]	14
2.9	Mathematical process models overview [Ise06]	16
2.10	RO system modeled in [BCC09]	21
3.1	Structure of the reverse osmosis plant	29
3.2	Abstract representation of the membrane module	30
3.3	Hydraulic network with the electronic-hydraulic analogy	36
3.4	Pressure values as a function of the concentration C_{ms} and the feed stream Q_f for a constant valve opening of 30 % (a) and 50 % (b)	39
3.5	Simulation of the RO system in open loop	46
3.6	Simulation of the RO system in open loop with fault f_1	49
3.7	Simulation of the RO system in open loop with fault f_2	50
3.8	Simulation of the RO system in open loop with fault f_3	51
4.1	Bipartite graph of system (4.1) and (4.2)	56
4.2	Different matchings for the bipartite graph in 4.1	59
4.3	Example graphs for the Hall's Marriage Theorem	60
4.4	Example for the Dulmage-Mendelsohn decomposition [BKL ⁺ 16]	61
4.5	Bipartite graph and over-constrained subgraphs of equation (4.18)	66
4.6	Bipartite graph of equation (4.20) before and after augmenting the matching	71
4.7	Steps of the simple algorithm for finding MSOs	74

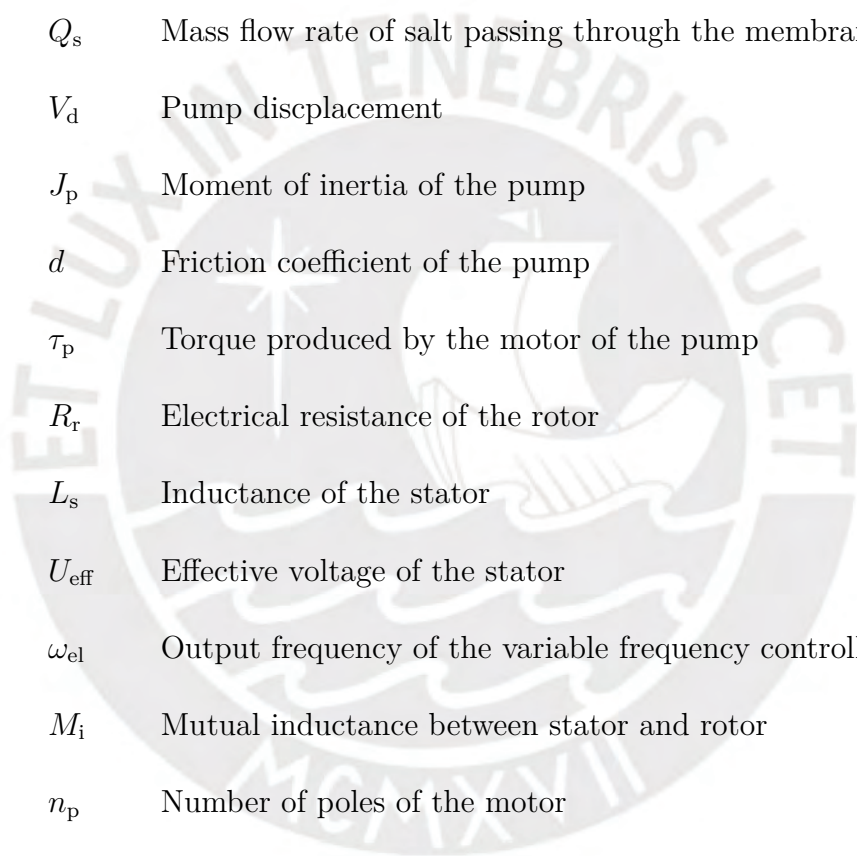
4.8	Example of building equivalence classes in a bipartite graph	74
5.1	Structure of the online fault diagnosis system	85
5.2	Output of the residual generators in the faultless case	91
5.3	Output of the residual generators when fault f_1 gets active	92
5.4	Output of the residual generators when fault f_2 gets active	94
5.5	Output of the residual generators when fault f_3 gets active	95
5.6	Output of the residual generators when fault f_4 gets active	95
5.7	Output of the residual generators when fault f_5 gets active	97
5.8	Output of the residual generators when fault f_6 gets active	97



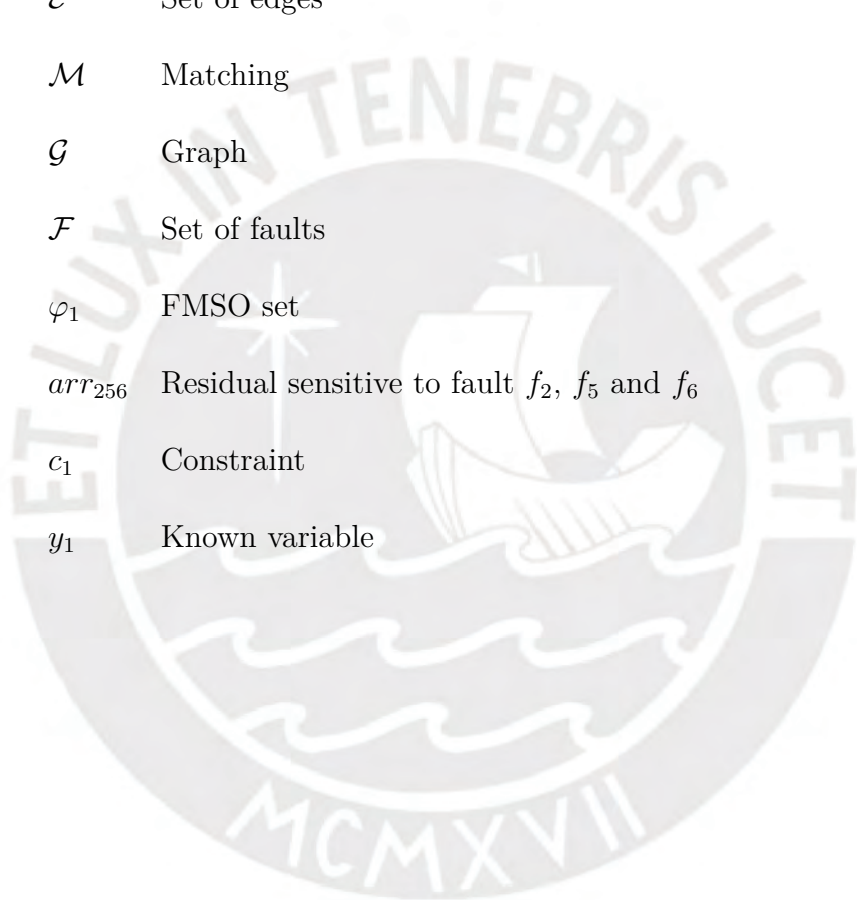
List of Symbols



f	Fault
ω_p	Rotational speed of the pump
ΔP	Pressure difference across the membrane
Q_p	Volumetric flow rate of the permeate stream
C_p	Salt concentration of the permeate stream (low)
V_b	Control volume on the brine side of the membrane module
V_p	Control volume on the permeate side of the membrane module
Q_f	Volumetric flow rate of the feed stream
Q_b	Volumetric flow rate of the brine stream
Q_m	Volumetric flow rate passing through the membrane
C_f	Salt concentration of the feed stream
C_b	Salt concentration of the brine stream
$\Delta\pi$	Osmotic pressure difference across the membrane
A_m	Surface area of the membrane
k_m	Solvent permeability of the membrane



\mathcal{R}	Ideal Gas constant
T	Temperature of the solvent
M_m	Molar mass of salt
C_{ms}	Salt concentration on the membrane surface at the brine side (high)
C_m	Salt concentration on the membrane surface at the permeate side (low)
Q_s	Mass flow rate of salt passing through the membrane
V_d	Pump displacement
J_p	Moment of inertia of the pump
d	Friction coefficient of the pump
τ_p	Torque produced by the motor of the pump
R_r	Electrical resistance of the rotor
L_s	Inductance of the stator
U_{eff}	Effective voltage of the stator
ω_{el}	Output frequency of the variable frequency controller
M_i	Mutual inductance between stator and rotor
n_p	Number of poles of the motor
ω_{el}^*	Nominal value of the supply frequency
u_p	Factor of the input frequency
α_v	Flow coefficient of the valve
A_v	Cross sectional area of the valve
ρ_b	Density of the brine stream
τ_v	Mechanical time constant of the valve



A_v^*	Nominal value of the cross sectional area of the valve
u_v^*	Opening factor of the valve
γ_p	Conductivity of the permeate stream
x_1	Unkown variable
\mathcal{V}	Set of vertices
\mathcal{E}	Set of edges
\mathcal{M}	Matching
\mathcal{G}	Graph
\mathcal{F}	Set of faults
φ_1	FMSO set
arr_{256}	Residual sensitive to fault f_2 , f_5 and f_6
c_1	Constraint
y_1	Known variable

List of Abbreviations

ARR Analytical Redundancy Relation

DAE Differential Algebraic system of Equations

DM Dulmage-Mendelsohn

FMSO Fault-driven Minimal Structurally Overdetermined

I/O Input/Output

MSF Multi-Stage Flash distillation

MSO Minimal Structurally Overdetermined

MTES Minimal Test Equation Support

PCA Principle Component Analysis

PSO Proper Structurally Overdetermined

RO Reverse Osmosis

SO Structurally Overdetermined

SOM Self Organizing Maps

STD Standard Deviation

Introduction

Water is one of the essential substances for making life possible on earth. Although most of the earth's surface is covered by water, only a small amount is drinkable for humans.

Population and economic growth increase the demand for drinking water. In addition to our responsible handling of the resource, existing technologies for water treatment must be improved, or new ones have to be developed to face water scarcity. These actions are necessary to maintain or even improve drinking water access for every human being in this world.

One of these water treatment technologies is reverse osmosis. With reverse osmosis, it is possible to reduce the concentration of foreign molecules in a liquid by pressing it through a semipermeable membrane. The results are a liquid with a high concentration of the molecules and a liquid with a lower concentration.

The main application for reverse osmosis is the desalination of water. This technology's usage is steadily increasing because the desalted water can be used for drinking or even for medical applications. It is more efficient than water purification through thermal processes because there is no phase change of the liquid.

Compared to other processes, the RO has the same disadvantage of maintenance, e.g., the membrane can clog and significantly reduce the process's efficiency. Thus, monitoring the process state is necessary to prevent losses through faults or even a breakdown of the plant.

Process monitoring done by a worker is expensive and requires a certain level of experience by the worker. A fault diagnosis system that monitors the process and notifies when a fault is active is a better solution. Such a system needs to be designed just once and then works without additional costs.

A common approach for developing a fault diagnosis system is through a mathematical model that describes the process behavior based on physical laws. Fault diagnosis via structural analysis exploits the profound knowledge provided by the mathematical model. It is, therefore, a reliable method to detect occurring faults in a system.



CHAPTER 2

State of the art

Hand in hand with the growing water scarcity on our planet grows the need for either improving existing technologies or discovering new ones for fresh water production. Even though water desalination had not been taken into account for drinking water production for a long time, since the last half of the last century its popularity for exactly this aim has been growing more and more.

In this chapter at first an introduction to the most popular technologies for water desalination is given. After that, the term fault diagnosis is defined and different approaches are summarized. Then follows an overview about recent works dealing with the fault diagnosis for the reverse osmosis process and the chapter finishes with the objectives of this work.

2.1 Water desalination

Desalination is defined as the process of removing salt from water. Normally, this process involves three different water streams with different salinity. The so called feed stream is processed into the brine stream and the permeate stream. The input stream, called feed stream, usually consists of brackish water which has a salinity between 0.5 - 30 grams per liter or sea water with a typical salinity in the range of 30 - 45 g/l. The product of the process is the permeate stream with a lower salinity than the feed stream. The aim of the product is to have drinking water quality which means less than 0.5 grams of salt per liter. Since the total amount of salt entering to the system with the feed stream cannot be reduced, the brine stream contains a higher amount of salt than the feed stream and is therefore seen as the waste product of the process. The maximum salinity of the brine stream is 390 g/l as this is the maximum solubility of NaCl in water with a temperature of 100 °C. [TSB⁺18; DW19]

The desalination is a process that occurs in nature without the need for an external trigger. Water evaporates from the sea and later comes back to earth as rain with a lower amount of salt. One of the first mentions of seawater desalination in humans history was made by Aristotle 320 BC. During the following centuries different techniques for water desalination were found. Most of them are copying the nature by evaporating salty water and condensing it on a surface like a fleece or a sponge. The need for water desalination in those times was rather small. The first practical use for water desalination occurred in the 16th century when sailors started to cross the Atlantic and faced drinking water scarcity on their ships. [BCA⁺15]

With the population growth throughout the last century the demand for drinking water has increased, which makes the water desalination not just interesting for sailors anymore. Hand in hand with the increasing demand goes the development of new technologies for water desalination. Figure 2.1 gives an overview of the existing technologies nowadays.

The main technologies used for drinking water production can be divided into two main groups. The thermal processes which are based on the principle of evaporating water and condensing it and the membrane driven technologies which do not include a phase change of the water. The latter can also be seen as filtration process.

There are alternative processes like freezing salty water. During the formation of ice crystals, dissolved salts are excluded and the obtained ice then can be defreezed to get less salty water which also includes a phase change just in the other “direction”. Even though this approach has a theoretical lower energy requirement than the thermal processes with heating

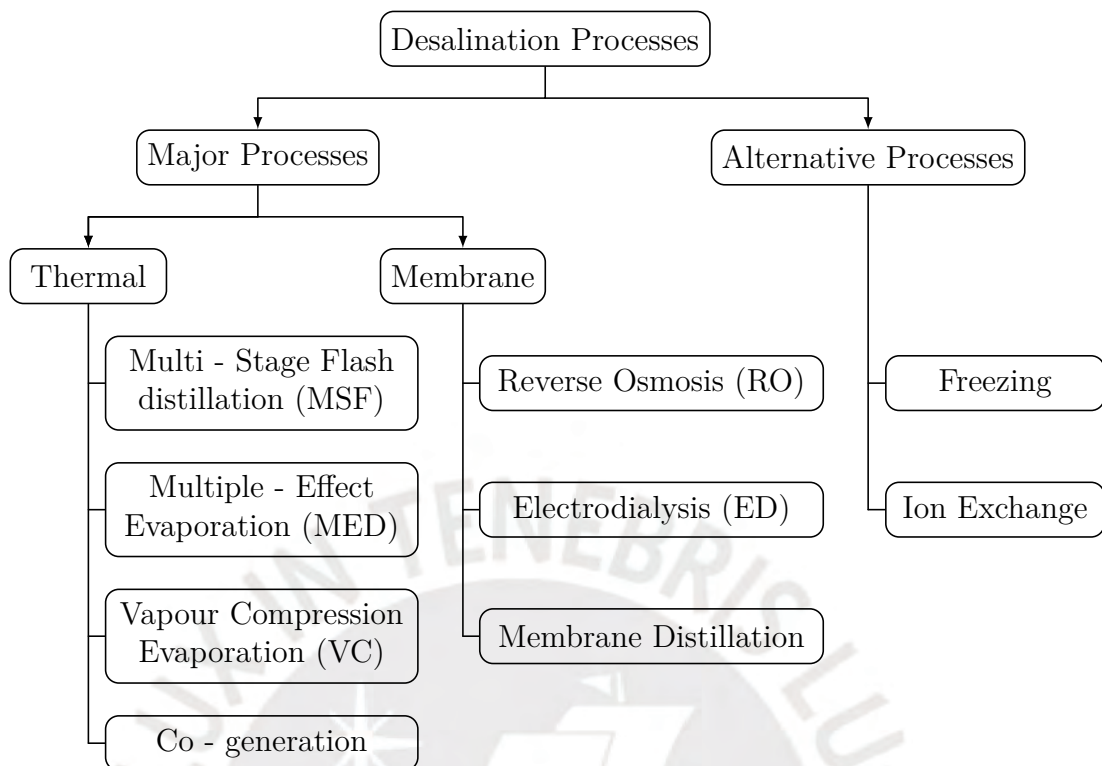


Fig. 2.1: Classification of water desalination processes [SR14]

involved, the technology does not get much attention. One reason for that might be the difficult handling of water and ice in automated processes.

Another alternative approach is to extract the salt via ion exchange. The salt dissolved in water splits into Na^+ and Cl^- ions. These ions are exchanged by H^+ or OH^- respectively which can be done until a complete demineralization of the NaCl solution is accomplished. A problem with this technology is that the solids which are capable of exchanging the ions need to be regenerated after a certain amount of extracted salts. The process therefore is not suitable to desalinate sea or brackish water because of the extensive costs. [SR14]

The aforementioned technologies have just a small percentage of the overall fresh water production through desalination. For drinking water production the most used technologies are the Multi - Stage Flash distillation (MSF) and the Reverse Osmosis (RO). [DW19] The advantages and disadvantages of each process will be discussed in the following sections.

2.1.1 Multi - Stage Flash distillation

The Multi - Stage Flash distillation (MSF) is the most popular desalination technology among the thermal processes. Until the end of the last century it was the most used technology for water desalination covering around 60% of the total world production of desalinated water. The reason for this is its robustness and capability to produce a large amount of fresh water. [DW19; Naj16]. A schematic of the process can be seen in figure 2.2.

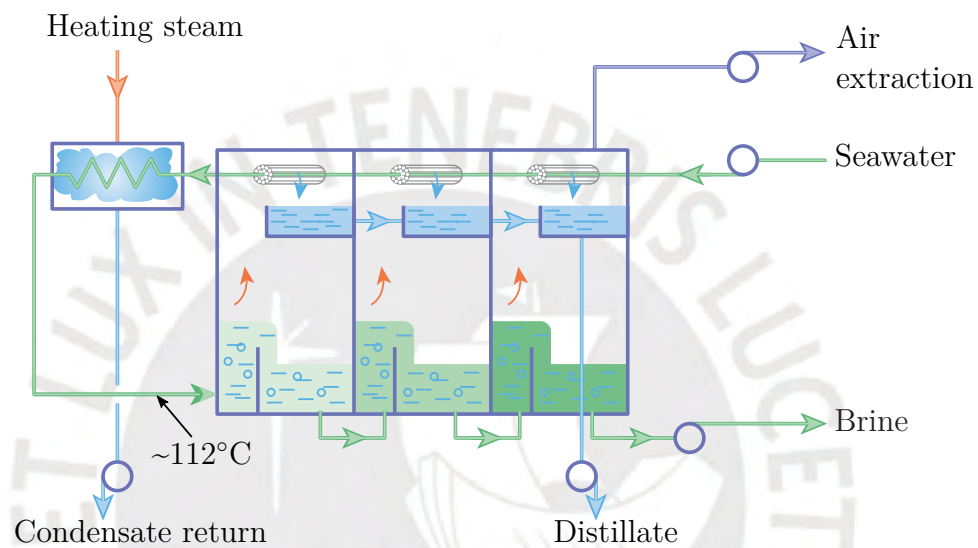


Fig. 2.2: Schematic of MSF process [Naj16]

The MSF process consists of multiple stages connected in series. In this case three stages are shown, typically there are 15 to 25 stages. Each stage is a chamber which contains the cycle of evaporating and condensing water. The seawater, when still cold, first flows through condensing coils in the upper part of each chamber. Afterwards it passes through a heat exchanger where it is heated up until it reaches the boiling temperature. The heated water then passes through every chamber, where in every chamber a part of it evaporates. To improve the evaporation process, within each chamber a lower ambient pressure is generated. The resulting steam condenses at the condensing coils and is then collected by a collecting tray underneath the coils. With this process between 20 - 30% of the feed water can be recovered as product water [SR14; DW19].

Some advantages and disadvantages of this technology are [SR14]:

- + MSF plants are relatively simple to construct and operate
- + The product water contains 2 - 10 ppm solids which is a high level of purification
- + The quality of the feed water is not as important as it is in the reverse osmosis technology
- It is considered as an energy intensive process
- More stages improve efficiency and increase water production but also raise the capital costs and operational complexity
- Operating the plant at higher temperature improves efficiency but causes dissolved salts like calcium sulphate (chalk) to cause mechanical problems like tube clogging

Even though the overall efficiency of the MSF plants can nowadays be improved through operating the plant with renewable energy, it lost its dominance in water production in favor of the reverse osmosis [DW19].

2.1.2 Reverse osmosis

Osmosis is the flow of water across a semipermeable membrane due to a solute concentration difference across the membrane. This difference produces a water flow through the membrane from the side with the lower concentration to the side with the higher concentration of molecules whereas most of the molecules are rejected by the membrane [CCE06]. Figure 2.3 shows a closed system containing a salt solution on one side of the membrane and pure water on the other side. Due to the concentration difference, water starts to flow through the membrane. Also a little amount of salt is passing the membrane as the rejection is not perfect. The salt stream goes in the opposite direction though. These flows take place until the hydraulic pressure generated by the different water levels is equal to the so called osmotic pressure ($\Delta\pi$). The osmotic pressure is determined through the concentration difference across the membrane and the temperature of the solution [EE02].

If a pressure $\Delta\pi$ greater than the osmotic pressure is applied to the solution with the higher concentration, the direction of the water flow changes but the direction of the salt flow doesn't. This phenomena is then called reverse osmosis. Figure 2.4 shows a closed system with a salt solution, the feed solution, on one side of the membrane and an empty volume on the other side. If now a pressure higher than the osmotic pressure of the solution is applied,

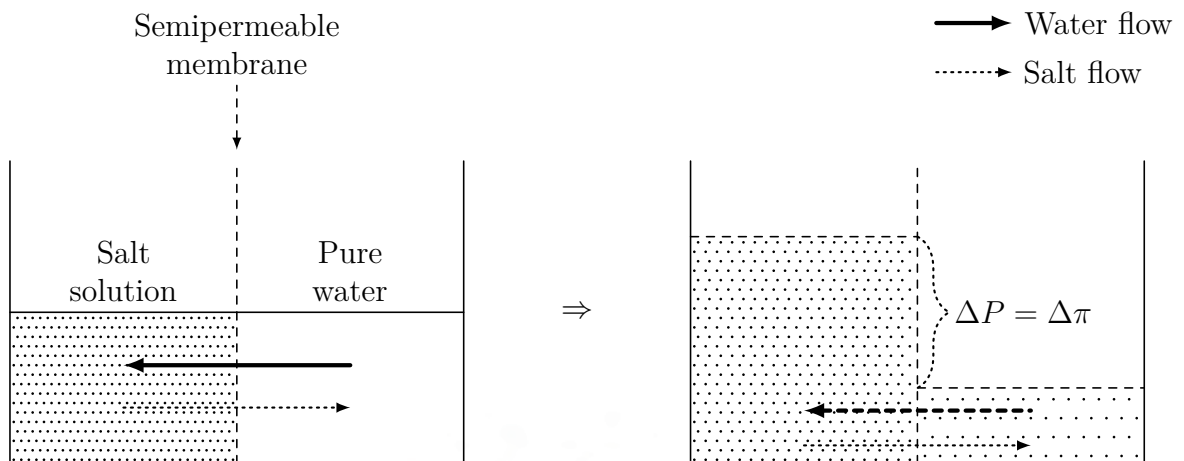


Fig. 2.3: Process of osmosis [EE02]

water starts to cross the membrane and also a little amount of salt. The permeate solution, which is the solution passing through the membrane, has a lower salt concentration than the feed solution. The concentration of the permeate solution depends on the concentration of the feed solution and the applied pressure. While the flow takes place, the concentration of the feed solution raises. The process stops when it reaches the equilibrium where the applied pressure equals the osmotic pressure [EE02]. This approach for reverse osmosis is not useful for water production as it is not continuous. To make it a continuous process an open system is needed, where the higher concentrated feed solution (brine) is substituted continuously. Therefore, a reverse osmosis plant has another setup as the one shown in figure 2.4 as shown in the next section.

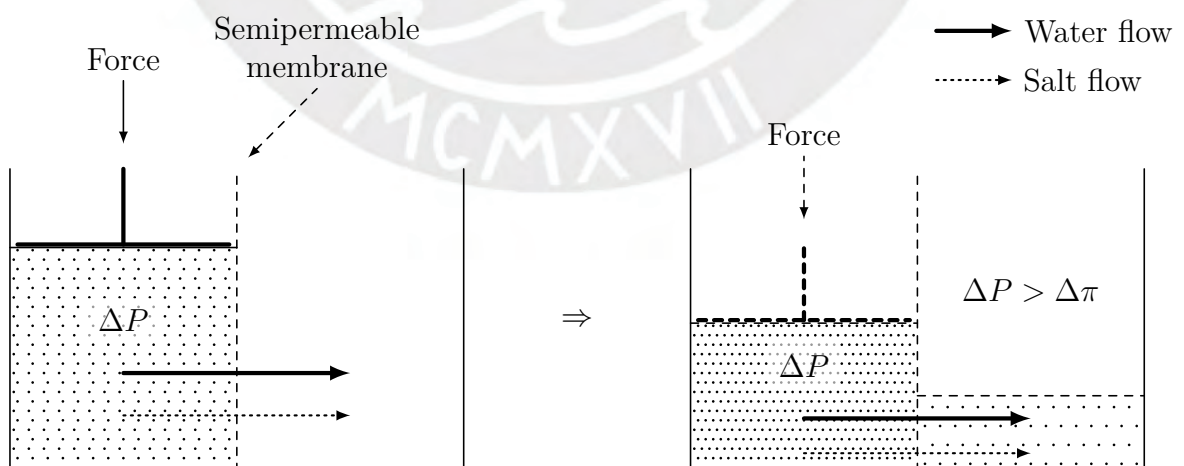


Fig. 2.4: Process of reverse osmosis [EE02]

2.2 Reverse osmosis plant

The usage of the reverse osmosis as a technology for industrial water desalination started with the first commercialization of a RO membrane module by Du Pont in 1967 [WCH⁺11]. In the following section the main parts of a RO plant will be discussed on the pilot plant, which this work is based on. The real plant is situated in the control and automation laboratory of the Pontificia Universidad Católica del Perú (PUCP). The setup of the pilot plant is shown in figure 2.5.

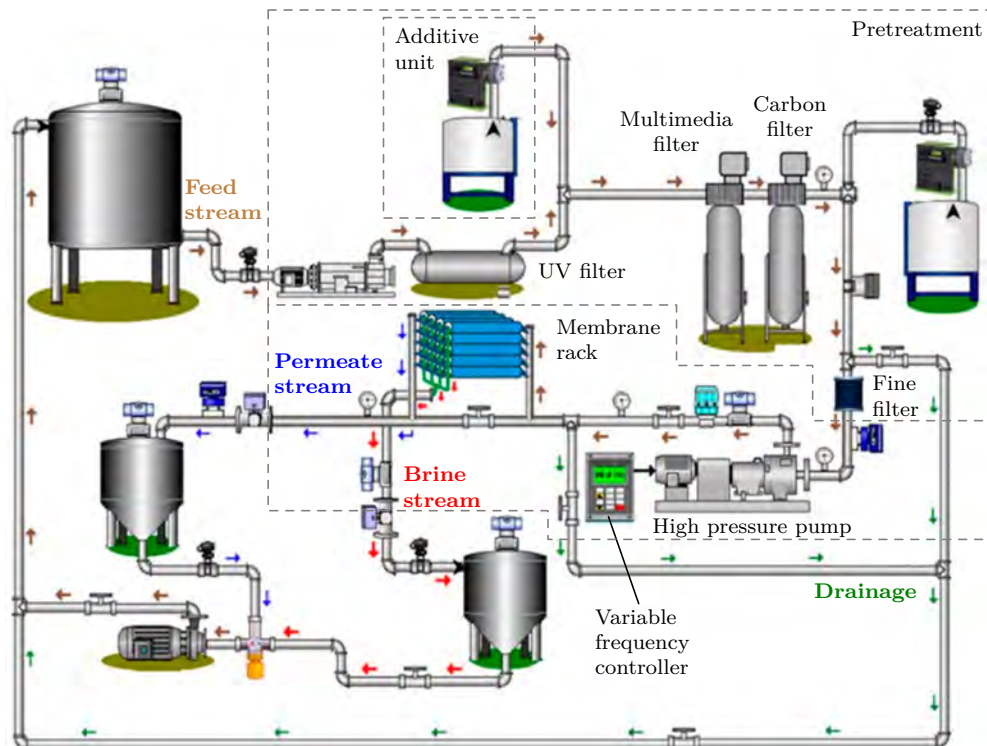


Fig. 2.5: Reverse osmosis pilot plant [RSP⁺19]

A RO plant usually consists of three parts. The pretreatment, RO process and posttreatment. The posttreatment, which is not shown in figure 2.5, depends on the required quality of the product water and will not be discussed any further in this work. An example for a posttreatment is the pH stabilization [WCH⁺11].

In addition, in figure 2.5 are shown the drainage and the different tanks for the water storage. For a production plant these parts are not needed, but for the pilot plant it is necessary because it is not connected directly to a salt water source. The water in this case is treated in a cycle of desalination and salinization. The feed water concentration can therefore be adjusted to the desired operating point of the plant.

Pretreatment

Before the raw feed water enters to the high pressure part, it undergoes a pretreatment. This pretreatment is necessary for all RO plants and is a disadvantage compared to the MSF process where this step is not mandatory. The pretreatment consists of the following steps. At first the water passes through a UV filter. The UV Filter eliminates biodegradable substances like plant materials in the raw water [FC17]. Afterwards, the water passes through some mechanical filters where suspended solids are extracted. Beside the filtering, the pretreatment consists of adding additives to the feed solution. The additives can be acid for adjusting the pH value and/or antiscalent or Chlorine. All the aforementioned measures have the aim to prevent the following problems taken from [STP+08].

1. Scaling is the deposition of dissolved metal salts on the membrane surface. To prevent this antiscalent additives are used.
2. Fouling is the deposition of solid particles on the membrane surface. Thus different mechanical filters are used.
3. Biofouling is the growth of bacteria on the membrane surface. The measures to prevent this are the UV filter and/or chlorine additives.

Even with the presence of all the measures, the occurrence of one of these problems cannot be excluded. They all lead to the same issue which is a clogged membrane. Even a partial clogged membrane is a problem as it reduces the efficiency of the plant. A clogged membrane can be flushed with a cleaning solution which in most of the cases requires the removal of the membrane module from the plant [Bak00].

Reverse osmosis

After the pretreatment of the feed stream, it enters to the high pressure pump which is driven by a variable frequency controller that allows to control the rotational speed of the pump. The required pressure is typically between 15 - 25 bar for brackish water and between 50 - 80 bar for sea water and is determined by the salinity of the feed water, the type of the membrane used for the desalination and the desired concentration and amount of the permeate stream.

Afterwards, the feed stream is separated into the permeate and the brine stream inside of the membrane rack. A proportional valve allows to control the brine stream which leaves the membrane module. The generated pressure is also affected by this valve as it will be

shown later in chapter 3 when the RO process is modeled. The membrane rack is the core of a RO plant.

It usually contains several membrane modules, depending on the size of the plant, which can be connected in parallel or series. In the parallel case an advantage is, that one of the modules can be removed or replaced without shutting down the plant first. The pilot plant provides two spiral wound membrane modules which can be connected in both ways. In this work only the parallel case is taken into account, because connecting the membrane modules in series usually requires additional pressure generation between the connected modules which cannot be provided.

Membrane module

There are two major types of membrane modules used for reverse osmosis plants: The hollow fiber and the spiral wound modules.

Hollow fiber membranes are comparable to tubes with an inner diameter of approximately $50\ \mu\text{m}$, where the outer wall of the tube acts as the membrane. A membrane module contains millions of fibers bundled together and folded in half so that all the fiber openings are on one side (see figure 2.6a). In the center of the fiber bundle is a tube with holes where the feed water enters. The feed water is then pressed through the fiber bundle and is separated into the brine and the permeate stream. The permeate stream is generated within the fibers where it is led to the outlet of the membrane module. The same happens with the brine stream after passing the fiber bundle.

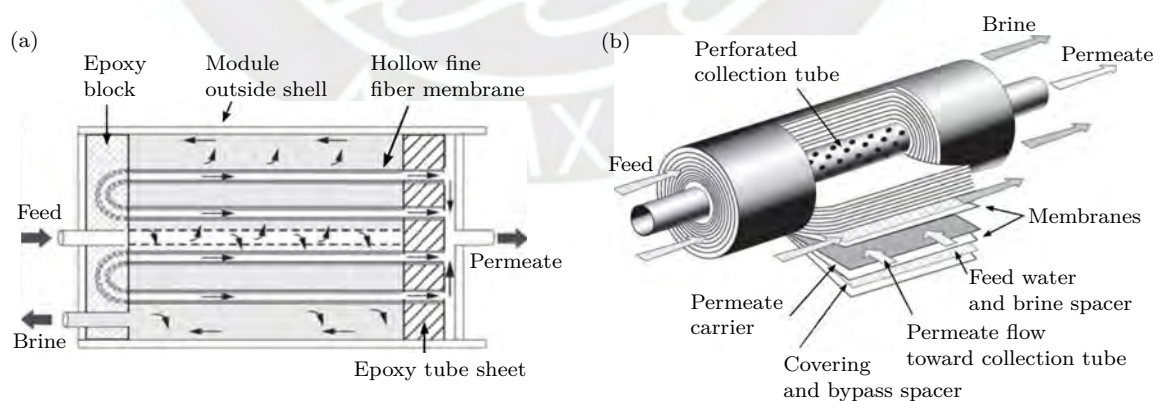


Fig. 2.6: Hollow fiber (a) and spiral wound (b) membrane module [Kuc15]

The spiral wound membrane module is the most used type for the reverse osmosis plants because of its simple production and high packaging density. In a spiral wound module

the membranes are not tubes but sheets placed on both sides of a permeate carrier. The permeate carrier is a nylon mesh that acts as a spacer between the two membrane sheets which form the permeate channel. This channel leads the permeate stream in a spiral form to the perforated collection tube in the center of the membrane module where it is led to the outlet. On the other side of the two membrane sheets are the feed/brine spacers which provide the channel for the feed/brine stream. In this order various layers can be achieved which in the end are wrapped around the collection tube (see figure 2.6b). [Kuc15]

The main difference between the two module types is the package density which is the effective membrane area per volume of the membrane module. In this case hollow fiber modules can reach higher densities because of their design. The design however has the disadvantage of “dead” areas, areas between the hollow fibers where water cannot flow properly, which makes it more sensitive to membrane fouling and difficult to clean. Thus hollow fiber modules need a higher quality of feed water to prevent fouling and scaling. [Kuc15]

Beside the different module designs exist various types of membrane material. Here the spiral wound module has the advantage that it is not restricted to a certain material. The most used membrane type nowadays are interfacial composite membranes because of their higher fluxes and higher salt rejection capability compared to the first invented cellulose acetate membranes. Composite membranes reach a salt rejection of around 99.5% whereas the typical rejection factor of a cellulose acetate membrane is between 98 - 99%. Although the difference of the rejection factors is small, it is important to know that a rejection of 99.3% or higher is required to produce potable water from seawater with just one stage of membrane modules. In addition to the better salt rejection, the water passage through the same area of the membrane is twice as high as with cellulose acetate membranes.

Reasons why cellulose acetate membranes are still in use are their cheap production in combination with a hollow fiber module, which leads to a high package density. Furthermore, because of the high tolerance against chlorine which is sometimes necessary to prevent biofouling. [Bak00]

2.3 Fault diagnosis

A fault in the technical sense is an incorrect deviation of one of the process states of a system. In a dynamical system this can be caused either by the change of the system structure e.g a blocking actuator, or by a parameter change which is caused for example through wear or damage. Occurring faults can lead to an interruption or even a breakdown of the process. [BKL⁺16]

2.3.1 System behavior

Regardless of the system model used for analysis, a system can be characterized through its inputs $u \in \mathcal{U} \subseteq \mathbb{R}^n$ and its outputs $y \in \mathcal{Y} \subseteq \mathbb{R}^m$. A pair (u, y) is called input/output (I/O) pair which are known values of the system at a certain time. Then the system behavior $\mathcal{B} \subseteq \mathcal{U} \times \mathcal{Y}$ is determined by all possible I/O-pairs which are consistent with the behavior of the system.

For a better understanding of fault diagnosis, a visualization of the sets is given in figure 2.7. The area in gray shows the set \mathcal{B} which is the model behavior for all I/O-pairs. In addition, point A and C are drawn into the space $\mathcal{U} \times \mathcal{Y}$. Point A belongs to the specific I/O-pair (u_A, y_A) and the point C to the pair (u_C, y_C) . The measured I/O-pair (u_A, y_A) belongs to the set \mathcal{B} and is therefore consistent with the model behavior. The point C does not belong to \mathcal{B} thus it can be assumed that the system is faulty when the pair (u_C, y_C) is measured. Note that the set \mathcal{B} does not need to be a fixed set within $\mathcal{U} \times \mathcal{Y}$. For dynamic systems figure 2.7 can be seen as a time slice, because the behavior \mathcal{B} also depends on the foregoing values of y and u . [BKL⁺16]

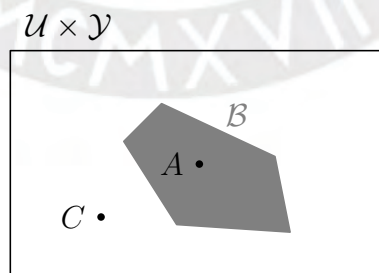


Fig. 2.7: Graphical illustration of the system behavior [BKL⁺16]

2.3.2 Consistency based fault diagnosis

The main idea of fault diagnosis is to find the fault f for a given I/O-pair. Usually f is not just limited to one fault but to a set of faults \mathcal{F} with $f \in \mathcal{F}$ where the element f_0 symbolizes the faultless case. More specific the diagnostic problem consists of the following steps:

The first step is the **fault detection**, which is to determine if a fault has occurred. If there is more than one fault, **fault isolation** takes place in the next step which means determining which fault $f \in \mathcal{F}$ has occurred. Furthermore, the fault diagnosis can be extended to determine the magnitude of the fault, so that further decisions for measures based on the magnitude can be made. All these tasks are usually made online, which means that the faults need to be detected while the plant is operating and in the best case as soon as possible to prevent further damage caused by a fault.

A graphical illustration for the difference of fault detection and isolation is given in figure 2.8 with the aforementioned notion of the system behavior. According to the influence of the fault on the system behavior, it creates a new set in the $\mathcal{U} \times \mathcal{Y}$ space. In figure 2.8 four additional sets are shown where each of the set belongs to a fault. The behavior \mathcal{B}_0 belongs to f_0 which is the faultless case, behavior \mathcal{B}_1 belongs to fault f_1 and so on. As explained before, a measured I/O-pair can now be assigned to a certain behavior. The point A belongs to the set \mathcal{B}_3 , but \mathcal{B}_3 is a subset of \mathcal{B}_0 . The fault f_3 is therefore not detectable as its behavior coincides with the normal behavior of the system.

Point B lays in the intersection of \mathcal{B}_1 and \mathcal{B}_2 , but outside of \mathcal{B}_0 . It can be determined that the system is faulty but not which fault is changing the behavior. Hence, fault f_1 and f_2 are detectable but not isolable at least when the measured I/O-pair belongs to the set $\mathcal{B}_1 \cap \mathcal{B}_2$. The last case includes point C which belongs just to the behavior \mathcal{B}_4 . \mathcal{B}_4 does not intersect with another set which makes f_4 a detectable and isolable fault. [BKL⁺16]

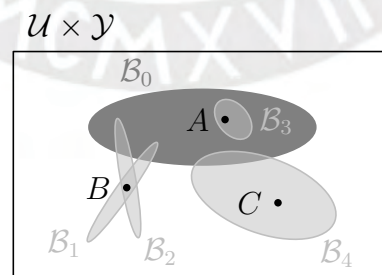


Fig. 2.8: Change of behavior for a faulty system [BKL⁺16]

It can be summarized that:

Definition 2.3.1. (Detectable fault) [BKL⁺16]

A fault $f \in \mathcal{F}$ is detectable if its behavior differs from the nominal behavior \mathcal{B}_0 of the system. Which holds when the faulty behavior $\mathcal{B}_f \not\subseteq \mathcal{B}_0$ and $\mathcal{B}_f \neq \emptyset$.

Definition 2.3.2. (Isolable fault) [BKL⁺16]

Two detectable faults $f_i, f_k \in \mathcal{F}$ with $i \neq k$ are isolable if their behavior $\mathcal{B}_i, \mathcal{B}_k$ do not intersect. Which holds when $\mathcal{B}_i \cap \mathcal{B}_k = \emptyset$.

These definitions conclude that fault detection is possible without any information about the behavior of the faulty plant, by evaluating if an I/O-pair is consistent with the nominal behavior or not. But without the knowledge of how a certain fault affects the system behavior, no fault isolation is possible. Furthermore, if a fault is neither detectable nor isolable, this does not mean that it is not possible at all. Taking into account other information of the system, which means a change of $\mathcal{U} \times \mathcal{Y}$, can result in a detectable or isolable fault. [BKL⁺16]

Consistency based diagnosis is a general idea which does not depend on the type of model used for describing the behavior of the system. The behavior can be described through rules like it is the case, for example, in spell checking. There the nominal behavior is described by a dictionary and a fault is present, when the word does not exist in the dictionary.

This work considers only mathematical systems with real-valued inputs and outputs, thus $\mathcal{U} \subseteq \mathbb{R}^n$ and $\mathcal{Y} \subseteq \mathbb{R}^m$. For this case there are several approaches which differ in the used model and the assumptions for the consistency based diagnosis.

2.3.3 Process models

A mathematical process model can be obtained by either theoretical or experimental modeling. An overview of the different models is given in figure 2.9.

Theoretical modeling, also called modeling by first principles, aims to describe the process behavior through the laws of nature. The resulting equations can be static or dynamic and of linear or non-linear type. If the process is time- and space-dependent, the behavior is expressed through partial differential equations. Often the space dependency can be neglected so that the system can be described through ordinary differential equations which

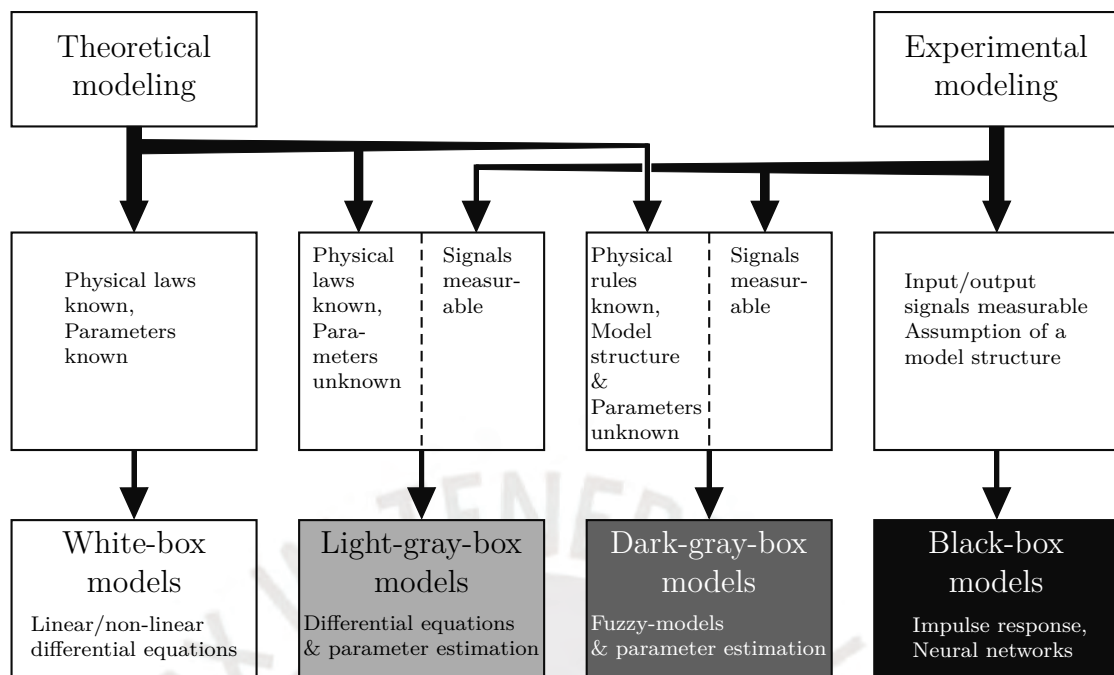


Fig. 2.9: Mathematical process models overview [Ise06]

expresses the system behavior as a function of time. The parameters of the equations can be time-dependent or time-independent. [Ise06]

If the parameters or its variation in time are known, the resulting model is called a white-box model. With a white box model the process can be simulated and analyzed without further measurements of the original process. This is useful for designing a process according to the change of its parameters or for designing control algorithms based on stability theories. [Ise06]

In some cases the physical laws of the process are available, but the parameters are not. In this case the model is called a light-gray-box model. Further information of the process can be obtained by measuring the process signals which can be used to estimate the parameters of the model.

If the modeling via physical laws is not possible or rather too complex, experimental modeling is used to determine a process model. For experimental modeling, also called identification, it is essential to have the real process signals. Depending on the available knowledge a dark-gray-box or a black-box model needs to be identified. An example of a dark-gray-box model is a fuzzy-model with certain known rules of the process, like if-then rules. The existing parameters also need to be estimated with the help of the process signals.

Black-box models are used if no further insights to the process are given except the measured signals. Then a model structure needs to be assumed and its parameters need to be estimated. Examples for this approach are neural networks or the identification through test signals like the impulse or step response. [Ise06]

2.3.4 Fault detection methods

Fault detection via state estimation

Fault detection via state estimation is an important field of fault detection via residuals. For this method a white-box or light-gray-box model is required. The main idea of this approach is to use the available model equations for estimating known/unknown variables of the process and compare them with the known/measured inputs and outputs. [Ise06]

A linear time-invariant system can be described via the state space model:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (2.1)$$

$$y(t) = Cx(t) \quad (2.2)$$

With $x \in \mathbb{R}^n$ the process states, $u \in \mathbb{R}^m$ the inputs of the system and $y \in \mathbb{R}^m$ the outputs of the system. The outputs y are measured at any time and the entries of the matrices A , B , and C are known. If the system is observable, the state vector x can be estimated with the equation [Ise06]:

$$\hat{\dot{x}}(t) = A\hat{x}(t) + Bu(t) + Le(t) \quad (2.3)$$

$$e(t) = y(t) - C\hat{x}(t) \quad (2.4)$$

Substituting the output error $e(t)$, the observer equation can be written as:

$$\hat{\dot{x}}(t) = [A - LC]\hat{x}(t) + Bu(t) + Ly(t) \quad (2.5)$$

With equation (2.2) and (2.5) the differential equation of the estimation error $e_x(t) = \hat{x}(t) - x(t)$ can be written as:

$$\dot{e}_x(t) = [A - LC]e_x(t) \quad (2.6)$$

Equation (2.6) shows that if $[A - LC]$ is a Hurwitz matrix, which can be accomplished by a proper design of the observer feedback matrix L . Then the error $e_x(t)$ converges to zero for any initial value of \hat{x}_0 . The error $e(t)$ in equation (2.2) also converges to zero and can be

used for fault detection. [Ise06]

For example, a system with multiplicative faults can be expressed as:

$$\dot{x}(t) = [A + \Delta A]x(t) + [B + \Delta B]u(t) \quad (2.7)$$

$$y(t) = [C + \Delta C]x(t) \quad (2.8)$$

Now the estimation error leads to:

$$\dot{e}_x(t) = [A - LC]e_x(t) + [\Delta A - L\Delta C]x(t) + \Delta Bu(t) \quad (2.9)$$

$$e(t) = Ce_x(t) + \Delta Cx(t) \quad (2.10)$$

The state and output error depend on the parameter changes of the matrices, which can be used for fault detection. Whether the parameter changes are detectable and isolable, depends on the particular case. [Ise06]

Another approach for state estimation is the Kalman filter which considers, unlike the before shown approach, unmeasurable disturbances.

Also the idea of state estimation can be extended to non-linear models which can lead to a more comprehensive analysis. Fault detection via state estimation requires a profound knowledge about the process in form of a mathematical model but provides provability and has the advantage that for designing the fault diagnosis system no process data is required, unlike the fault detection with black-box-models like neural networks.

Fault detection via neural networks

Fault detection via neural networks is based on geometric classification. The neural network aims to reduce a given error function, which can be for example [Ise06]:

$$\min V(\mathcal{B}_j) = \min_{\forall \mathcal{B}_j \in \mathcal{B}_{\mathcal{F}}} \sqrt{\|s - \mathcal{B}_j\|^2} \quad (2.11)$$

The error function (2.11) is the Euclidean distance between the input signals s and a given reference behavior \mathcal{B}_j of the process. The measured process signals denoted as s are the input of the neural network. s is not just restricted to currently measured values. To take into account the dynamics of a system, previously measured values can be included in s . The goal of the neural network is to determine which set \mathcal{B}_j reduces the error function $V(\mathcal{B}_j)$, thus it classifies the input values s . If the solution is $\mathcal{B}_j \neq \mathcal{B}_0$, then a fault is active. [Ise06]

The output of the neural network are the expectation values $E\{t_j/s\}$, where t_j is the target value and indicates how much s coincides with a certain fault. Beside the process signals s , the output also depends on the weights w which are determined through the training of the neural network. Hence one output $y_j(s, w)$ of the neural network can be written as [Ise06]:

$$y_j(s, w) = E\{t_j/s\} = \int t_j p(t_j|s) dt_j \quad (2.12)$$

$p(t_j|s)$ is the probability of occurrence of t_j for a given s . If the neural network has one output per fault and the target value t_j is chosen to be binary like [Ise06]:

$$t_j(s) = \begin{cases} 1, & \text{if } s \text{ belongs to fault } f_j \\ 0, & \text{otherwise} \end{cases} \quad (2.13)$$

Then equation (2.12) can be simplified to [Ise06]:

$$y_j(s, w) = P(f_j|s) \quad (2.14)$$

Where $P(f_j|s)$ is the probability that s belongs to fault f_j .

Fault detection via neural networks is a growing field in fault detection. [Ise06] assumes that more than half of the fault diagnosis systems rely on neural networks, which is mostly because of its simple implementation through the efficient tools available nowadays.

One disadvantage of using neural networks for fault detection is that there is no proof of convergence of the algorithm. For the detection of faults via a neural network, it needs to be trained with data. During the learning phase the weights w of the neural network are determined. In the best case, w converges to a final value where equation (2.11) has its minimum. There is no proof for that because there is no further knowledge about the process except the data obtained through measurements. Another disadvantage is that the data for training requires a working plant. And for fault isolation all faults must be first caused and the neural network needs to be trained to detect them, which can be difficult depending on the fault.

Other possible fault detection methods

Beside the fault detection methods via process models, it is possible to detect faults via signal models. For example the vibrations of a machine can be described through the superposition of oscillations.

$$y(t) = \sum_{k=1}^n y_{0k} \sin(\omega_k t + \varphi_k) \quad (2.15)$$

Then, a fault can then be detected by evaluating the amplitude spectrum of the vibrations of the machine. [Ise06]

Another method for fault detection is the limit checking of the measured signals. A process is said to be faulty if the measured signal $y(t)$ is out of predetermined bounds:

$$y(t) < y_{min} \text{ or } y_{max} < y(t) \quad (2.16)$$

This method is also applied to trend checking by evaluating the time derivative $\dot{y}(t)$ instead of $y(t)$.

Similar to fault detection via state estimation is the fault detection through parameter estimation. In this case not the deviation of the process states is analyzed but the deviation of the system's parameters. This type of fault detection also requires a precise knowledge about the process model. [Ise06]

Fault detection via structural analysis, which will be applied in this work, also uses the mathematical model. With this approach the model structure is analyzed to find overdetermined systems of equations within the model. Based on this, residuals which indicate if a fault is active or not can be deduced. This method will be explained with more detail in chapter 4.

2.4 Recent works

2.4.1 Existing mathematical models of reverse osmosis plants

For fault diagnosis via structural analysis a mathematical model in form of equations is essential. In literature one can find various approaches for the modeling of the reverse osmosis which will be discussed in the following.

[Kim17] gives an overview of the different equations, used to model the reverse osmosis. The author focuses on modeling the static behavior of the fluids and concentrations of the different streams separated by the membrane. The mass balance is used to model the streams and concentrations. The water and salt diffusion through the membrane are seen as independent streams and are modeled via the Spiegler-Kedem hyper filtration model which was developed in 1966 [Kim17; SK66]. Furthermore, a static space-independent model of the concentration-polarization phenomena is given. A more detailed modeling of this phenomena requires taking into account its dependency of space and time which results in a partial differential equation. A more detailed model is given by the same author in [KH05].

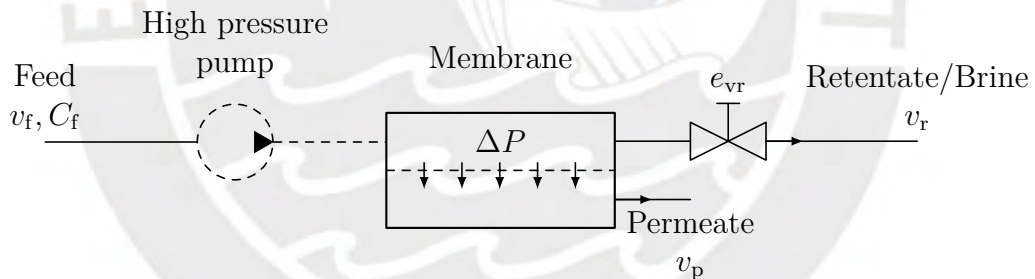


Fig. 2.10: RO system modeled in [BCC09]

A dynamic model is introduced in [BCC09]. This model contains one process state which is the brine stream velocity controlled by a valve. For modeling the salt-rejection and membrane stream also the Spiegler-Kedem model is used. The structure of the model is shown in figure 2.10.

Taking the energy balance around the brine valve leads to the following equation [BCC09]:

$$\dot{v}_r = \frac{P_{\text{sys}} A_p}{\rho V} - \frac{1}{2} \left(\frac{A_p e_{\text{vr}} v_r^2}{V} \right) \quad (2.17)$$

Where:

- v_r Brine stream velocity
- P_{sys} System pressure
- A_p Cross sectional area of the pipe
- V System volume
- ρ Density of the fluid
- e_{vr} Valve resistance

This equation can be obtained by considering the water within the system accelerated by the system pressure. The system pressure P_{sys} is calculated by taking into account the steady state balance of the streams under the conditions that all streams have the same density and all pipes have the same diameter which leads to [BCC09]:

$$0 = v_f - v_r - v_p \quad (2.18)$$

With v_f the feed stream velocity and the velocity v_p of the permeate stream passing through the membrane. v_p can also be expressed with the Spiegler-Kedem model [BCC09]:

$$v_p = \frac{K_m A_m}{\rho A_p} (P_{\text{sys}} - \Delta\pi) \quad (2.19)$$

Where:

- K_m Membrane overall mass transfer coefficient
- A_m Membrane area
- $\Delta\pi$ Osmotic pressure

With equation (2.18) and (2.19) the system pressure can be substituted with:

$$P_{\text{sys}} = \frac{\rho A_p}{A_m K_m} (v_f - v_r) + \Delta\pi \quad (2.20)$$

The osmotic pressure $\Delta\pi$ is calculated with:

$$\begin{aligned}\Delta\pi &= \delta C_{\text{eff}} (T + 273) \\ C_{\text{eff}} &= C_f \left\{ a + (1 - a) \left[(1 - R) + R \left(\frac{v_f}{v_r} \right) \right] \right\}\end{aligned}\quad (2.21)$$

Where:

- C_{eff} Effective average concentration at the membrane surface (feed side)
- C_f Concentration feed stream
- δ Weighting coefficient
- a Weighting coefficient
- R Fractional salt rejection of the membrane

Substituting equation (2.20) and (2.21) in (2.17) gives the model used for controlling the reverse osmosis plant. The valve resistance is the manipulated input and is related to the open percentage O_v of the valve through the static equation:

$$O_v = \mu \ln(e_{vr}) + \varphi \quad (2.22)$$

With the experimentally determined values μ and φ . For controlling the plant, two independent controllers are used. A PI-controller is used to keep the system pressure P_{sys} at a given set-point. The controller adjusts the frequency of a variable frequency controller which powers the motor of the pump. This actually changes the feed flow rate which affects the system pressure [BCC09]. An explicit model for this behavior is not given.

The second control loop is driven by a non-linear controller and regulates the brine stream velocity v_r through the valve resistance e_{vr} . The feed stream velocity v_f is seen as an independent variable which is taken into account for the design of the controller and is used as a measured variable [BCC09]. A connection between the two loops is not given.

The system modeled in [BCC09] is equal to the pilot plant in this work. The problem is that essential couplings between variables are not modeled, for example the linkage between the feed stream, system pressure and the pump. Furthermore, only the feed stream concentration is examined, but other interesting variables like the permeate concentration or the permeate stream are not modeled and not controlled.

A dynamic model which considers more variables and couplings is given in [GKB07]. Here the salt and water streams are modeled independently which allows to model the dynamic of concentrations via the mass balance. The system modeled in [GKB07] contains an energy recovery system to heat up the feed stream. This temperature can be controlled and is used

as an input of the system. A higher temperature of the feed stream allows a higher amount of water passing through the membrane while the concentration of the permeate stream decreases. [GKB07]

Another input of the system modeled in [GKB07] is the brine stream valve. The feed flow rate though is considered constant and is not controlled by the pump.

The hollow-fiber membrane module modeled in [GKB07] leads to a special form of the Spiegler-Kedem equation. The model does not have the same structure as the pilot plant used in this work, but many ideas can be taken, which will be explained in more detail in chapter 3.

2.4.2 Existing fault diagnosis systems for reverse osmosis plants

Data based approaches

Most of the works found about fault diagnosis of a reverse osmosis plant are based on measurements from a real plant. The obtained data is then used to estimate a system model. In [PGB⁺14] the data from the following sensors are obtained in a certain range and with the respective standard deviation (STD). The evaluated sensors are the feed flow rate, feed conductivity, feed pressure, permeate flow rate, permeate conductivity, brine flow rate, brine conductivity and the pressure of the brine stream.

With this data a model is estimated based on support vector regression. The estimated model is then used to predict the sensor data. The predicted sensor data is compared and evaluated with self-organizing maps (SOM) which do not require a model for each plant sensor. This approach is similar to the neural network approach explained in section 2.4.2 and leads to an accurate fault-detection for sensor deviations of greater than 10%. [PGB⁺14]

Another data based approach can be found in [GFP11], where the fault detection is realized by using the principal component analysis (PCA). With this approach the data is not evaluated directly but rather the principle components. Therefore, the data vectors y are stored in a matrix X and the covariance matrix S is calculated with:

$$S = \frac{1}{K-1} X^T X \quad (2.23)$$

Where K is the number of samples stored in X . Furthermore, the matrix X has to be normalized to zero mean and unit variance. The principle components are then the eigenvectors of S which belong to the biggest eigenvalues of the covariance matrix. These

eigenvectors can then be used for fault detection by comparing them with the eigenvectors of the covariance matrix of a faulty system. [GFP11]

Model based approaches

A robust model based fault diagnosis system is introduced in [SCS11]. The fault detection is based on a bond graph model which models the RO plant via analogies between the different fields of physics. The connections between the different parts of the plant have either the characteristics of a resistance, capacitance, inductance or a supply. With these elements it is possible to model the dynamic behavior of the plant, by interconnecting each part. The interconnections are based on the plant and its behavior. Therefore, some knowledge about the process is required and the value of each element needs to be determined experimentally. Based on this model the structure of the process can be analyzed to find residuals for fault detection. [SCS11]

In [MBC⁺08] an observer based approach is presented to detect valve faults. The dynamical model is similar to the one presented in section 2.4.1 from [BCC09] in the form of differential equations. The model is used to estimate the brine stream velocity which is then compared with the measured value. A fault is then said to be active, if the residual, which is the absolute value ϵ of the difference of the estimated value $\hat{y}(t)$ and the measured value $y(t)$, is bigger than a given threshold for longer than a given time, as shown in equation (2.24).

$$\epsilon = |y(t) - \hat{y}(t)| \quad (2.24)$$

If a fault is active, a different control strategy is used to guarantee closed loop stability. [MBC⁺08]

Another approach is presented in [Sot16] which was made with the same plant used in this work. The evaluated faults are:

1. Faulty conductivity sensor
2. Faulty flow rate sensor
3. Faulty high pressure pump
4. Faulty temperature sensor
5. Membrane wear

6. Membrane damage
7. Membrane fouling
8. Faulty additive pump pH

The used model comprises 15 equations but only one dynamic equation which describes the dynamic behavior of the brine stream concentration. The model assumes that the system pressure can be set directly by the high pressure pump. This assumption neglects the hydraulic circuit which is comprised by the brine stream valve and the membrane module. Unfortunately, the faults are only considered by which equation they affect, but not how they appear within the equation. Therefore, it cannot be determined if the fault is multiplicative or additive and neither how the fault affects the calculated residuals.

Furthermore, the fault of the membrane damage is not simulated correctly because it is simulated with a sudden decrease of the membrane area which is similar to a sudden clogging/fouling of the membrane. This can be also seen in the results, since all the faults are detectable, but the faults 3, 5 and 6 are not isolable. As stated above, fault 5 is the membrane wear (simulated by changing the membrane coefficients) and fault 6 the membrane damage.

The fault detection is realized by analytical redundancy relations (ARR) as it will be done in this work. An ARR is a relation, which depends on the measured model outputs and inputs. If no fault is active, the relation is fulfilled, else it is not fulfilled. The exact definition of an ARR is given in section 4.5.1. ARRs are deduced from an overdetermined system of equations which can be found via structural analysis, therefore various algorithms exist in literature. In [Sot16] two algorithms are used for finding these overdetermined system of equations. First the ranking algorithm or also called matching algorithm is used, which results in 10 evaluated sets of equations, hence 10 ARRs. This is not the maximum number of overdetermined sets which can be found, but sufficient to detect the 8 evaluated faults. The selection of the considered ARRs depends on which ARR detects which faults and will be discussed later in this work in chapter 4 and 5.

Beside the Ranking algorithm, an algorithm for determining so called Minimal Test Equation Supports (MTES) is applied, which is presented in [KÅF10]. The algorithm finds 8 sets of equations which can be used to deduce an ARR. The problem is that a proper analysis cannot be made as the faults are not introduced properly, furthermore just two of the 8 sets are used to deduce an ARR.

In this work, the fault diagnosis system will also be based on structural analysis. However, the algorithm utilized will be different from the one used in [Sot16]. Furthermore, the

mathematical model will be extended, and the faults will be modeled as well. The extended model and the different algorithms lead to different results than those obtained in [Sot16]. Furthermore, assuming that just one fault at a time occurs, for detecting and isolating eight faults, only four ARRs would be necessary. This assumption will also be considered when selecting the overdetermined systems of equations.



Mathematical modeling of the reverse osmosis plant

Many applications in the field of control and automation rely on a mathematical model that describes the process behavior. The model can be of different nature depending on the information needed. For example, control applications require the dynamic behavior of a system, which can be described through transfer functions or differential equations. Whereas for optimization purposes the static coupling between the process states can be sufficient. For a fault diagnosis system every kind of information is valuable, because a change in both static and dynamic behavior can provide information about the occurring faults.

In the following chapter the mathematical model of the reverse osmosis plant is developed, which provides the base for the diagnosis system. After an introduction to the plant and its structure which has to be modeled, the plant is divided into three parts. The parts consist of the membrane module, the high pressure pump and the hydraulic circuit which are coupled among each other.

Afterwards the evaluated faults are determined and finally, the simulation results of the plant and its faults are shown.

3.1 The model structure

For developing the model of the RO plant, it is first decomposed into smaller subsystems. This is a common method for modeling huge systems, since it simplifies the modeling process. The pre-treatment, which includes additional filtering and pH-adjustment, is not modeled, because in the interesting range of operation they do not have an effect on the desalination process [Kuc15]. An approach for modeling the pre-treatment of a RO plant is given in [STP⁺08]. The structure of the plant which is modeled, is shown in figure 3.1.

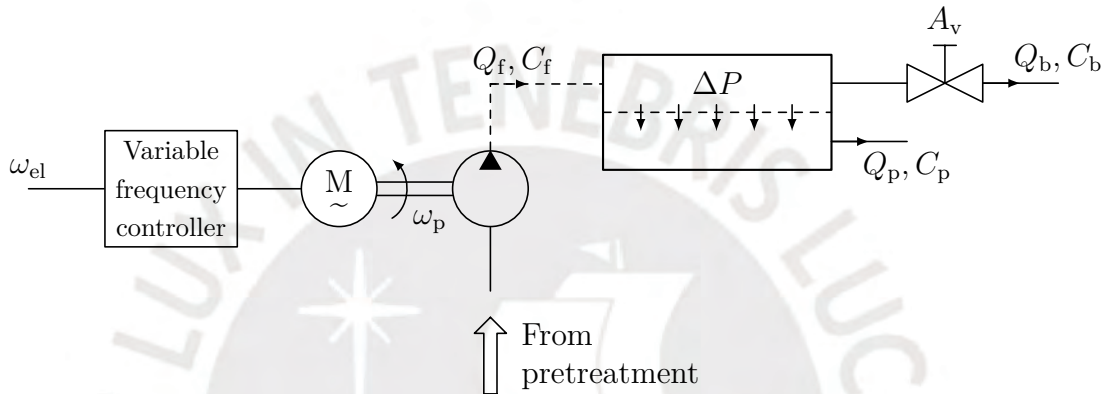


Fig. 3.1: Structure of the reverse osmosis plant

As mentioned, the RO plant is divided into two subsystems which are part of a hydraulic system. One subsystem comprises the high-pressure pump driven by an electrical motor. The motor is connected to a variable frequency controller which makes it possible to control the rotational speed ω_p of the pump. The first input of the whole system is the frequency set point ω_{el} which is processed by the variable frequency controller. During normal operation of the plant, ω_{el} is set by a control algorithm.[RSP⁺19]

The second subsystem is the membrane module, where the actual desalination takes place. The pump generates the feed stream Q_f with its concentration C_f . Within the membrane module, the feed stream is separated into the permeate stream Q_p passing through the membrane and the brine stream leaving the membrane module through the brine valve. It is also possible to control the brine valve and thus to change its cross sectional area A_v . The set point of the cross sectional area A_v is the second input of the whole system. Together with the membrane, the brine valve acts like a resistance to the feed stream. This is the reason why a pressure ΔP within the membrane module is generated. This pressure causes the reverse osmosis to take place at the membrane where the feed stream is separated into the permeate stream Q_p with the concentration C_p and the brine stream Q_b with the

concentration C_b . The pressure drop ΔP across the membrane generates the permeate stream Q_p with a salt concentration C_p . Q_p and C_p are the product of the RO plant and they are the outputs which have to be controlled during operation.

The two subsystems together with the brine valve build an hydraulic system which determines the coupling between the two subsystems. [GKB07; RSP⁺19]

3.2 The membrane module

Since the first use of RO, many different membrane materials have been developed to improve the RO process. Improvements of the membrane material are not just aimed at a more efficient desalination but also to prevent membrane fouling. The considered plant has two spiral wound modules, which can be connected either in parallel or serial. In this work the case of parallel connection is modeled.

In terms of thermodynamics the membrane module can be split into two control volumes, shown gray and dotted in figure 3.2. V_b is the volume on the brine side of the membrane module. This is the side where the salt concentration and the pressure are high and V_p is the control volume on the side where the concentration and pressure are low. For the balance equations taken around the control volumes, all flows are considered as incompressible. Furthermore, the influence of the salt concentration on the density of the solution is neglected. Hence, it is assumed that all flows have the same density.

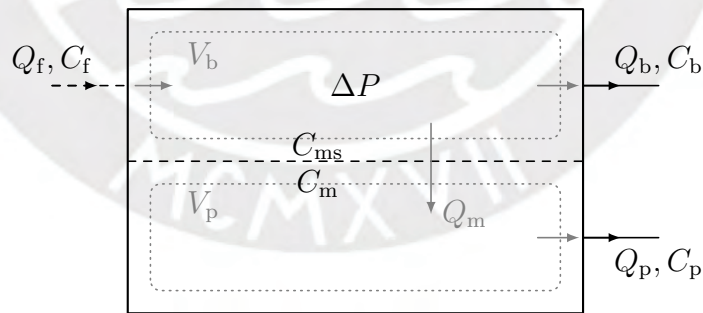


Fig. 3.2: Abstract representation of the membrane module

For control volume V_b then holds the equation [GKB07]:

$$Q_f = Q_m + Q_b \quad (3.1)$$

Where:

- Q_f Volumetric flow rate of the feed stream
- Q_b Volumetric flow rate of the brine stream
- Q_m Volumetric flow rate passing through the membrane

Furthermore, the differential mass balance for the salt streams around V_b leads to [GKB07]:

$$\frac{d(C_b V_b)}{dt} = Q_f C_f - Q_b C_b - Q_m C_m \quad (3.2)$$

Where:

- C_f Salt concentration of the feed stream
- C_b Salt concentration of the brine stream (high)

The control volume V_b is constant so that equation (3.2) can be simplified to:

$$\dot{C}_b = \frac{1}{V_b} (Q_f C_f - C_b Q_b - Q_m C_m) \quad (3.3)$$

For the control volume V_p , the solvent balance with the permeate stream Q_p and the membrane stream Q_m is:

$$Q_p = Q_m \quad (3.4)$$

The differential solute balance taken around the constant control volume V_p leads to:

$$\dot{C}_p = \frac{1}{V_p} (Q_m C_m - Q_p C_p) \quad (3.5)$$

The two volumes V_b and V_p are coupled through the semipermeable membrane which is modeled in the next section.

3.2.1 Mathematical model of the membrane

It is assumed, that the water (solvent) and salt (solute) pass through the membrane independently from each other. Thus the solvent (volumetric) flow Q_m passing through the membrane can be defined as [Kim17]:

$$Q_m = k_m A_m (\Delta P - \Delta \pi) \quad (3.6)$$

Where:

- $\Delta \pi$ Osmotic pressure difference across the membrane
- A_m Surface area of the membrane
- k_m Solvent permeability of the membrane
- ΔP Pressure difference across the membrane

Equation (3.6) shows that a higher pressure drop across the membrane produces a higher flow through the membrane.

The osmotic pressure difference $\Delta \pi$ which acts like a resistance, is defined as [Kim17]:

$$\Delta \pi = \frac{\mathcal{R}T}{M_m} (C_{ms} - C_m) \quad (3.7)$$

With:

- \mathcal{R} Ideal Gas constant
- T Temperature of the solvent
- M_m Molar mass of salt
- C_{ms} Salt concentration on the membrane surface at the brine side (high)
- C_m Salt concentration on the membrane surface at the permeate side (low)

This equation is also known as Van't Hoff equation and describes the relation between the two concentrations on either side of the membrane and the osmotic pressure. If the osmotic pressure is less than the pressure drop ΔP , the plant is operating in reverse osmosis mode. If $\Delta \pi$ is greater than ΔP , Q_m is negative which means that the osmosis takes place. This happens for example, when the high pressure pump is turned off suddenly.

In a similar way as the solvent flow Q_m , the solute flow (mass) Q_s passes through the membrane [Kim17]:

$$Q_s = k_s A_m (C_{ms} - C_m) \quad (3.8)$$

Where k_s is the solute permeability of the membrane.

With equation (3.6) and (3.8) the concentration C_m can be expressed as the ratio of solute flow to solvent flow [Kim17]:

$$C_m = \frac{Q_s}{Q_m} = \frac{k_s(C_{ms} - C_m)}{k_m(\Delta P - \Delta\pi)} \quad (3.9)$$

Substituting $\Delta\pi$ from equation (3.7) in (3.9) leads to,

$$C_m = \frac{k_s(C_{ms} - C_m)}{k_m(\Delta P - \frac{RT}{M_m}(C_{ms} - C_m))} \quad (3.10)$$

which defines the concentration C_m in an implicit way as it appears on both sides of the equation.

Nevertheless, equation (3.10) can be transformed to the quadratic form:

$$C_m^2 + C_m(\alpha + \beta\Delta P - C_{ms}) - \alpha C_{ms} = 0 \quad (3.11)$$

With:

$$\alpha = \frac{k_s M_m}{k_m RT} \quad \text{and} \quad \beta = \frac{M_m}{RT}$$

The solution is a root of the quadratic function (3.11):

$$\begin{aligned} C_m &= -\frac{1}{2}(\alpha + \beta\Delta P - C_{ms}) + \sqrt{\frac{1}{4}(\alpha + \beta\Delta P - C_{ms})^2 + \alpha C_{ms}} \\ &= \frac{1}{2} \left(C_{ms} - \beta\Delta P - \alpha + \sqrt{\beta^2\Delta P^2 - 2\beta\Delta P(C_{ms} - \alpha) + (C_{ms} + \alpha)^2} \right) \end{aligned} \quad (3.12)$$

Note that the quadratic equation of C_m in (3.11) has two mathematical solutions, but the negative solution is not feasible, because the concentration needs to be positive.

With equation (3.12) it is possible to determine the concentration C_m as a function of the pressure ΔP and the concentration C_{ms} on the membrane surface.

With equation (3.12) and (3.7) the membrane stream Q_m can also be expressed as a function of the pressure ΔP and the concentration on the membrane surface C_{ms} .

$$Q_m = \frac{k_m A_m}{2\beta} \left(\beta\Delta P - C_{ms} - \alpha + \sqrt{\beta^2\Delta P^2 - 2\beta\Delta P(C_{ms} - \alpha) + (C_{ms} + \alpha)^2} \right) \quad (3.13)$$

The salt concentration C_{ms} differs from the concentration C_b because of the concentration polarization. This phenomenon occurs because of the salt rejection of the membrane, which

leads to an increase of the salt concentration on the membrane's surface.

A description of C_{ms} is taken from [EE02] as:

$$C_{ms} = \frac{1}{2}(C_f + C_b) \quad (3.14)$$

Equation (3.14) is a simple approach to calculate the concentration on the membrane's surface.

Examining it closely, the concentration polarization is a two dimensional problem, because the concentration raises in the direction of the membrane's surface and also in the direction of the exit of the brine stream, which flows tangential to the surface of the membrane. Modeling the concentration polarization in a more detailed way leads to non-linear partial differential equations which are difficult to include in a fault diagnosis system. More detailed approaches are given in [KH05], [Bun86] or [RKS⁺13]. But even though (3.14) is a simple calculation of C_{ms} it indirectly considers, that C_{ms} depends on the membrane stream Q_m and the brine stream Q_b as they are coupled in equation (3.3).

After modeling the subsystems of the membrane module, the high pressure pump is modeled in the next section.

3.3 The high pressure pump module

For modeling the high pressure pump, the module is split into an electrical part which comprises the electrical motor controlled by a variable frequency converter and the mechanical part which is the pump itself.

The feed stream Q_f , generated by the pump, can be expressed as a function of the pump displacement V_d and the rotational speed n in rev/s or ω_p in rad/s of the pump:

$$Q_f = V_d n = \frac{V_d}{2\pi} \omega_p \quad (3.15)$$

The pump displacement V_d is the volume of the fluid, that the pump delivers per revolution.

The hydraulic power $\Delta P \cdot Q_f$ is equal to the mechanical power $M_p \cdot \omega_p$, therefore it holds that:

$$\Delta P Q_f = M_p \omega_p = \Delta P V_d n = M_p 2\pi n \quad (3.16)$$

Then the torque M_p generated by the pump can be expressed as a function of the pressure of the system with [Wat17]:

$$M_p = \Delta P \frac{V_d}{2\pi} \quad (3.17)$$

With Newton's law the differential equation for the rotational speed of the pump can be obtained as:

$$J_p \dot{\omega}_p = -\Delta P \frac{V_d}{2\pi} - d\omega_p + \tau_p \quad (3.18)$$

Where:

- J_p Moment of inertia of the pump
- d Friction coefficient of the pump
- τ_p Torque produced by the motor of the pump

A simplified model of the torque produced by the asynchronous motor is described in [Ise08]. This model neglects the time constants of the electrical part of the motor which is a valid simplification, as the mechanical part with ω_p and the concentrations C_b and C_p act in a range of seconds whereas the current within the motor reacts in the range of milliseconds. Thus, τ_p can be expressed as:

$$\tau_p = \frac{3n_p M_i^2 U_{\text{eff}}^2}{2R_r L_s^2 \omega_{\text{el}}} \left(1 - \frac{\omega_p n_p}{\omega_{\text{el}}}\right) \quad (3.19)$$

Where:

- R_r Electrical resistance of the rotor
- L_s Inductance of the stator
- U_{eff} Effective voltage of the stator
- ω_{el} Output frequency of the variable frequency controller
(Is equal to its set point frequency)
- M_i Mutual inductance between stator and rotor
- n_p Number of poles of the motor

The frequency converter which drives the motor has a so called "volts per hertz" mode. In this mode the ratio $U_{\text{eff}}/\omega_{\text{el}}$ is kept constant so that equation (3.19) can be simplified to:

$$\tau_p = c(-n_p \omega_p + \omega_{\text{el}}^* u_p) \quad (3.20)$$

With:

$$c = \frac{3n_p}{2R_r} \left(\frac{M U_{\text{eff}}}{L_s \omega_{\text{el}}}\right)^2$$

As ω_{el} is limited to a certain range, it is replaced by $\omega_{\text{el}}^* u_p$. ω_{el}^* is the nominal value of the supply frequency and u_p is limited to the range of 0 to 1.

The subsystem high pressure pump is modeled with the equations (3.15),(3.18) and (3.20), which are connected with the membrane module as shown in the following section.

3.4 Modeling the hydraulic circuit

The hydraulic circuit describes the interaction between the high pressure pump and the brine valve in order to generate the pressure drop ΔP across the membrane. In literature it is assumed, that either the pressure ΔP is constant [BCC09] or it is seen as input for the system and thus can be set independently as in [Sot16]. In the case of a reverse osmosis plant the last assumption is not valid, as the hydraulic system does not contain any pressure vessels. Moreover, the high pressure pump does not produce a pressure but a stream, more exactly the feed stream Q_f , as defined in equation (3.15).

For another visualization of the hydraulic circuit, the electronic-hydraulic analogy can be used as shown in figure 3.3. With this analogy the pressure drop is equivalent to the electric

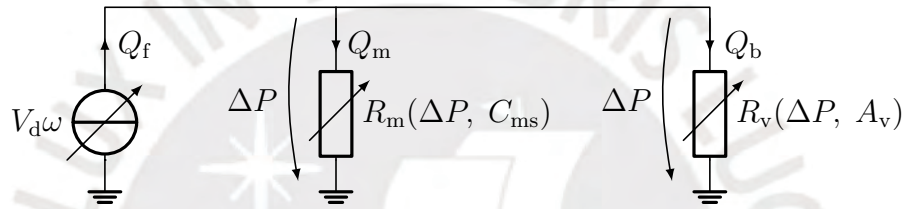


Fig. 3.3: Hydraulic network with the electronic-hydraulic analogy

potential and the volume flow rate corresponds to the electric current. The hydraulic pump can be seen as an adjustable current source. The current adjustment in this case is made by changing the rotational speed of the pump. The faster the motor rotates, the more flow the pump produces. The membrane and the valve act like two non-linear resistors in parallel. Thus, the pressure drop ΔP is generated by the flow streaming through the corresponding resistance.

The brine stream going through the valve is expressed as follows [Wat17]:

$$Q_b = \alpha_v \sqrt{\frac{2}{\rho_b}} A_v \sqrt{\Delta P} \quad (3.21)$$

Where:

- α_v Flow coefficient of the valve
- A_v Cross sectional area of the valve
- ρ_b Density of the brine stream

Furthermore, the valve has a big mechanical time constant, as it is not designed to open and close fast. Therefore, the dynamics of the valve can be described as:

$$\dot{A}_v = \frac{1}{\tau_v}(-A_v + A_v u_v) \quad (3.22)$$

Where:

- τ_v Mechanical time constant of the valve
- A_v^* Nominal value of the cross sectional area of the valve
- u_v^* Opening factor of the valve

u_v ranges from 0 to 1, where 1 means fully opened and 0 fully closed.

With equation (3.21) and (3.13) the streams Q_b and Q_m can be expressed in terms of the pressure ΔP . Substituting all the flows in equation (3.1) couples the three streams in the hydraulic circuit which is equivalent to Kirchoff's first law in terms of hydraulics. The stream Q_f does not depend on ΔP , it is therefore not substituted. Then it holds:

$$Q_f = \nu A_v \sqrt{\Delta P} + \frac{k_m A_m}{2\beta} \left(\beta \Delta P - C_{ms} - \alpha + \sqrt{\beta^2 \Delta P^2 - 2\beta \Delta P (C_{ms} - \alpha) + (C_{ms} + \alpha)^2} \right) \quad (3.23)$$

With $\nu = \alpha_v \sqrt{\frac{2}{\rho_b}}$

Through equation (3.23) the system pressure ΔP can be calculated according to the feed stream, the concentration on the membrane surface and the opening distance of the valve. But (3.23) is an implicit function of ΔP and to get the explicit function $\Delta P(Q_f, C_{ms}, A_v)$ is not straightforward.

Nevertheless, an analytical solution can be found with the substitution $\sqrt{\Delta P} = z$. Then the equation (3.23) can be brought to the form of a cubic polynomial:

$$0 = z^3 + a_2 z^2 + a_1 z + a_0 \quad (3.24)$$

with:

$$a_2 = \frac{\nu A_v}{k_m A_m} - \frac{Q_f}{\nu A_v} - \frac{\alpha k_m A_m}{\beta \nu A_v}$$

$$a_1 = -\frac{C_{ms} + \alpha}{\beta} - \frac{2Q_f}{k_m A_m}$$

$$a_0 = \frac{Q_f^2}{k_m A_m \nu A_v} + \frac{Q_f (C_{ms} + \alpha)}{\beta \nu A_v}$$

A general solution for a cubic polynomial can be found in [Con15] and [Sol]. At first the discriminant Δ needs to be calculated with:

$$p = a_1 - \frac{a_2^2}{3} \quad (3.25)$$

$$q = \frac{2a_2^3}{27} - \frac{a_2a_1}{3} + a_0 \quad (3.26)$$

$$\Delta = \frac{q^2}{4} + \frac{p^3}{27} \quad (3.27)$$

There are three possible cases, which determines the solution:

$$\text{Case 1 : } \Delta > 0 \quad (3.28)$$

$$\text{Case 2 : } \Delta = 0 \quad (3.29)$$

$$\text{Case 3 : } \Delta < 0 \quad (3.30)$$

If $\Delta > 0$, the equation (3.24) has only one real solution and two imaginary ones. For the case $\Delta = 0$, repeated roots are obtained which means that one unique real root and two roots with the same value which can be either complex or real. For the third case $\Delta < 0$, three different real roots are obtained. [Con15]

With the values of table 3.1 and within a range of interest for C_{ms} , A_v and $Q_f > 0$, the discriminant leads to $\Delta < 0$ and therefore, equation (3.24) has three real solutions. A solution for ΔP which then solves the equation (3.23) is:

$$\Delta P = \left[\frac{2}{\sqrt{3}} \sqrt{-p} \sin \left(\frac{1}{3} \arcsin \left(\frac{3\sqrt{3}q}{2(\sqrt{-p})^3} \right) \right) - \frac{a_2}{3} \right]^2 \quad (3.31)$$

The other solutions of equation (3.24) are irrelevant because not only ΔP , which is always positive because it is squared, but also z needs to be positive. Because of the transformation $z < 0$, it might be a viable solution as it still leads to $\Delta P > 0$, but this would lead to a negative brine stream through the valve which should never be avoided under normal conditions. The other solution is also positive but does not solve the equation (3.23), since the substitution introduces new zeros.

Figure 3.4 shows the values of the pressure ΔP as a function of the membrane surface concentration C_{ms} (10 - 50 kg m⁻³) and the feed stream Q_f (0 - 50 L min⁻¹) for two different constant values (30 and 50%) of the brine valve opening. In both cases the function is monotonic increasing with rising values of the feed stream and the concentration on the membrane surface. This can be explained by the electronic-hydraulic analogy: when the current (in this case the feed stream Q_f) through a resistor raises, the voltage drop

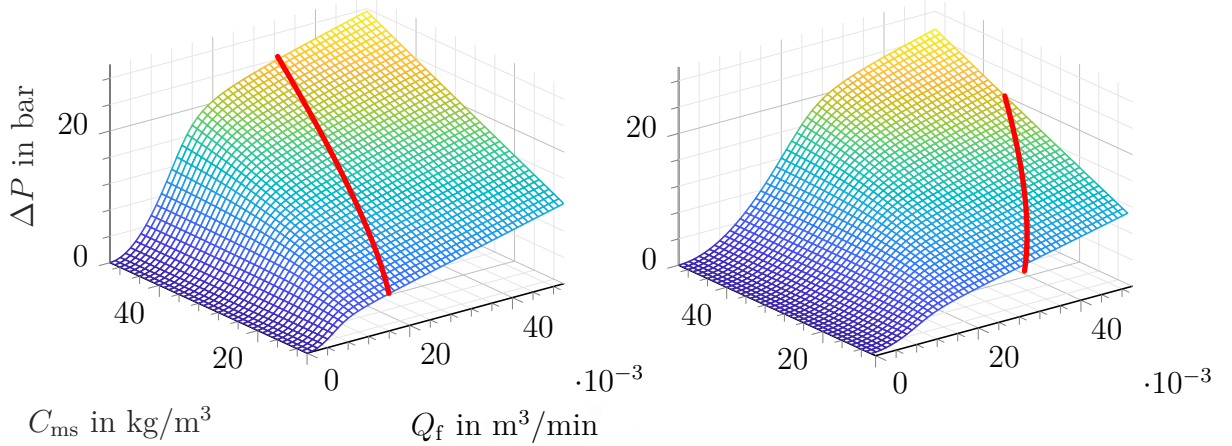


Fig. 3.4: Pressure values as a function of the concentration C_{ms} and the feed stream Q_f for a constant valve opening of 30 % (a) and 50 % (b)

(respectively, the pressure drop) raises too.

The increase of the concentration C_{ms} is equivalent to the raise of the resistance the feed stream needs to pass through, therefore the pressure also needs to raise. The same happens when the brine valve opening is decreased, which leads to a higher resistance for the stream. Therefore, the pressure increases more than if the valve is opened further. The red line in figure 3.4 shows where the two streams Q_m and Q_b are equal, thus $Q_f = 2 \cdot Q_b = 2 \cdot Q_m$ which leads to:

$$C_{ms} = \frac{Q_f^2 \beta + 2Q_f \alpha k_m A_m}{4\mu^2 A_v^2} - \frac{Q_f \beta}{2k_m A_m} - \alpha \quad (3.32)$$

On the right side of the red line the stream Q_m passing through the membrane is higher than the stream passing through the brine valve.

A more closed the brine valve leads to a higher product stream but leads also to a smaller brine stream Q_b . Aforementioned, due to the concentration polarization the concentration C_{ms} also depends on the brine stream. With a smaller brine stream the salts rejected by the membrane do not leave the membrane module in the same amount as with a higher brine stream, which leads to a higher salt concentration on the membrane surface.

It might not be necessary to find the explicit expression of the pressure for fault diagnosis, but it is useful for the simulation of the plant. Although equation (3.23) could be solved numerically, this leads to the problem that it requires an initial value every time the equation needs to be solved and it is less reliable and slower as an explicit calculation.

Nevertheless, an explicit expression for the pressure is necessary if a state space model for controlling purposes is required. Since the pressure ΔP is not a state, but depends on some states, ΔP needs to be replaced in every equation it occurs.

3.5 Measured and known variables

Based on the measured and known variables of the model, the fault diagnosis system is designed. In this work the following variables are measured (outputs) or known (inputs, parameters).

Measurements:

$$y_1 = pH$$

$$y_2 = Q_p$$

$$y_3 = Q_b$$

$$y_4 = \Delta P$$

$$y_5 = \gamma_p$$

$$y_6 = \gamma_f$$

Inputs:

$$y_7 = u_p$$

$$y_8 = u_v$$

With γ_p the conductivity of the permeate stream and γ_f the conductivity of the feed stream. These two variables are introduced to calculate the respective salt concentration as these cannot be measured directly. The relation between the concentration and the conductivity is given as [Sot16]:

$$\gamma_x = b_{x0}(C_x - C_x^*) - b_{x1}(pH - pH^*) + \gamma_x^* \quad (3.33)$$

With the constants b_{x0} and b_{x1} which need to be determined around the operating points C_x^* , pH^* and γ_x^* .

Furthermore all the static parameters of the model are assumed to be known. Summarized the behavior of the RO plant is described by a Differential Algebraic system of Equations (DAE) with the 22 equations shown in (3.36) - (3.53). The explicit expression for the pressure is not taken into account as it is not necessary for the fault diagnosis via structural analysis as mentioned in section 3.4. However, it appears as a boundary through the equations (3.36), (3.40) and (3.46).

Furthermore, some modifications are made, with the aim to reduce numerical errors. This is due to the fact that the values of some variables have a certain range. For example, the volume flow Q_f has a maximum value of 551/min, due to the limit of the high pressure pump. This value corresponds to $9.166 \cdot 10^{-4} \text{ m}^3/\text{s}$ in SI-units. On the other hand, the pressure can reach up to 20 bar which is equal to $20 \cdot 10^5 \text{ Pa}$ in SI-units. To minimize numerical errors caused by the huge difference of the values, the pressure is calculated in bar and all flows in m^3/min or kg/min respectively. Furthermore, the cross sectional area of the brine valve

is calculated in cm^2 . To accomplish this, the parameters of the model are changed to the according unit and the brine stream is changed to:

$$Q_b = \nu A_v \sqrt{\Delta P}, \quad \nu = 0.06 \alpha_v \sqrt{\frac{2}{\rho_b}} \quad (3.34)$$

Furthermore, the hydraulic torque from equation (3.17) is changed to:

$$M_p = \Delta P \frac{5 \cdot 10^4 V_d}{\pi} \quad (3.35)$$

The resulting parameters taken for the simulation are shown in table 3.1. The given constants need to be placed in the model in the given unit which sometimes differs from the SI-unit. Note that the shown parameters are an estimation of the real parameters of the plant.

Table 3.1: Parameter values taken for simulation

$M_m = 0.05844 \text{ kg mol}^{-1}$	$\rho_b = 1 \text{ kg l}^{-1}$	$C_f = 10 \text{ kg m}^{-3}$	$n_p = 1$
$V_b = 53.24 \times 10^{-3} \text{ m}^3$	$A_m = 40 \text{ m}^2$	$b_{p1} = b_{f1} = 0.05 \text{ S m}^{-1}$	$\alpha_v = 0.04$
$k_s = 12.0 \times 10^{-6} \text{ m min}^{-1}$	$A_v^* = 3.66 \text{ cm}^2$	$V_p = 35.49 \times 10^{-3} \text{ m}^3$	$pH^* = 7.0$
$c = 0.2 \text{ kg m}^2 \text{ s}^{-1} \text{ rad}^{-1}$	$J_p = 0.2 \text{ kg m}^2$	$\gamma_p^* = \gamma_f^* = 3.0 \text{ S m}^{-1}$	$\tau_v = 1.5 \text{ s}^{-1}$
$V_d = 13.04 \times 10^{-6} \text{ m}^3/\text{rev}$	$\omega_{el}^* = 377 \text{ rad s}^{-1}$	$C_p^* = C_f^* = 10 \text{ kg m}^{-3}$	$T = 298 \text{ K}$
$b_{p0} = b_{f0} = 0.1 \text{ S m}^2 \text{ kg}^{-1}$	$d = 0.002 \text{ N m s}$	$k_m = 12.0 \times 10^{-5} \text{ m min}^{-1} \text{ bar}^{-1}$	
$\mathcal{R} = 8.314472 \times 10^{-5} \text{ m}^3 \text{ bar mol}^{-1} \text{ K}^{-1}$			

List of equations:

$$Q_f = Q_m + Q_b \quad (3.36)$$

$$\dot{C}_b = \frac{1}{V_b}(Q_f C_f - C_b Q_b - Q_m C_m) \quad (3.37)$$

$$Q_m = Q_p \quad (3.38)$$

$$\dot{C}_p = \frac{1}{V_p}(Q_m C_m - Q_p C_p) \quad (3.39)$$

$$Q_m = k_m A_m (\Delta P - \frac{1}{\beta}(C_{ms} - C_m)) \quad (3.40)$$

$$C_m = \frac{1}{2} \left(C_{ms} - \beta \Delta P - \alpha + \sqrt{\beta^2 \Delta P^2 - 2\beta \Delta P (C_{ms} - \alpha) + (C_{ms} + \alpha)^2} \right) \quad (3.41)$$

$$C_{ms} = \frac{1}{2}(C_f + C_b) \quad (3.42)$$

$$Q_f = \frac{30V_d}{\pi} \omega_p \quad (3.43)$$

$$\dot{\omega}_p = \frac{1}{J_p} \left(-\Delta P \frac{5 \cdot 10^4 V_d}{\pi} - d\omega_p + \tau_p \right) \quad (3.44)$$

$$\tau_p = c(-n_p \omega_p + \omega_{el}^* u_p) \quad (3.45)$$

$$Q_b = \nu A_v \sqrt{\Delta P} \quad (3.46)$$

$$\dot{A}_v = \frac{1}{\tau_v} (-A_v + A_v^* u_v) \quad (3.47)$$

$$\gamma_p = b_{p0}(C_p - C_p^*) - b_{p1}(pH - pH^*) + \gamma_p^* \quad (3.48)$$

$$\gamma_f = b_{f0}(C_f - C_f^*) - b_{f1}(pH - pH^*) + \gamma_f^* \quad (3.49)$$

Measurements:

$$y_1 = pH \quad y_4 = \Delta P \quad (3.50)$$

$$y_2 = Q_p \quad y_5 = \gamma_p \quad (3.51)$$

$$y_3 = Q_b \quad y_6 = \gamma_f \quad (3.52)$$

Inputs:

$$y_7 = u_p \quad y_8 = u_v \quad (3.53)$$

3.6 Faults

In the following section the faults considered in this work and the modeling is presented. Beside the behavior of the fault like abrupt, drift-like or with interrupts, faults are usually divided into three different types [BKL⁺16]:

Plant faults change the dynamical I/O properties of the system, e.g. wear or increased friction.

Sensor faults do not affect the plant's properties but the sensor values do not coincide with the actual values of the system. As a sensor fault an offset or a trend is considered but not measurement noise.

Actuator faults do not affect the plant's properties but the influence on the plant is interrupted or different.

Each of these fault types can be modeled via two different approaches, namely as an additive or a multiplicative fault.

3.6.1 Multiplicative and additive faults

Multiplicative faults

Multiplicative faults are connected to a known parameter. The fault can be the change of a parameter expressed with $\Delta a(t) = f(t)$. This affects an equation in the following manner [Ise06]:

$$y(t) = (a + f(t))U(t) \quad (3.54)$$

Detecting the multiplicative fault depends on the variable $U(t)$, as the equation can never be solved for $f(t)$ if $U(t) = 0$.

Additive faults

Additive faults are independent from a variable and appear as additional terms in the equations like [Ise06]:

$$y(t) = aU(t) + f(t) \quad (3.55)$$

Unlike multiplicative faults, detecting additive faults does not depend on another variable as in this case for every value of $U(t)$ the equation can be solved for $f(t)$.

3.6.2 Faults of the reverse osmosis plant

Leakage of the feed stream f_1

A leakage can be caused by a worn gasket or a hole in a tube caused by oxidation. This leads to a decrease of the feed stream which can be described with:

$$Q_f = \frac{30V_d}{\pi}\omega_p - f_1 \quad (3.56)$$

This type of fault occurs suddenly and is therefore simulated as a step. Even though in this case f_1 is an additive fault, it depends on the variable ω_p , which is because Q_f needs to be greater than 0.

Membrane fouling f_2

Membrane fouling reduces the membrane area A_m .

$$Q_m = k_m(A_m - f_2)\left(\Delta P - \frac{\mathcal{R}T}{M_m}(C_{ms} - C_m)\right) \quad (3.57)$$

Membrane fouling happens gradually and is therefore a drift-like fault. It will be simulated with a ramp function.

Motor torque fault f_3

The torque of the motor can be affected through an external force accelerating or braking the rotor of the motor.

$$\tau_p = c(-n_p\omega_p + \omega_{el}^*u_p) - f_3 \quad (3.58)$$

This type of fault can happen suddenly or drift-like. For example, if a screw within the motor gets loose and falls into the rotating parts, it causes an abrupt braking. A drift-like occurrence could be the wear of a ball bearing which is also covered by this fault even though it would be dependent on ω_p . As long as the motor turns, this can be also detected.

Valve fault f_4

A valve fault would be a blockage or a displacement of the valve spool.

$$\dot{A}_v = \frac{1}{\tau_v}(-A_v + A_v^*u_v + f_4) \quad (3.59)$$

A valve displacement is an abrupt fault which is simulated by a step.

Sensor fault of Q_p f_5

A sensor offset of the permeate stream can be modeled as:

$$y_2 = Q_p + f_5 \quad (3.60)$$

Sensor fault of γ_p f_6

A sensor offset of the conductivity sensor of the permeate stream can be modeled as:

$$y_5 = \gamma_p + f_6 \quad (3.61)$$

The two sensor faults can be of any type, abrupt, drift-like or sporadically with interrupts. In this work only the abrupt case is simulated.

These are all faults which will be considered for the fault diagnosis. There can be added more faults, but this will lead to the problem that in the end they might not be isolable from each other anymore. Note that a fault can comprise several errors as mentioned for the fault f_3 . It is therefore difficult to say what exactly caused the fault.

3.7 Model evaluation and simulation

The model is simulated in Simulink in open loop with the initial values $[C_{b0}, C_{p0}, \omega_{p0}, A_{v0}] = [C_f, C_f, 0, 0.3A_v^*]$, C_f is kept constant at 10 kg m^{-3} and just the known variables are shown which are $y_1 - y_8$.

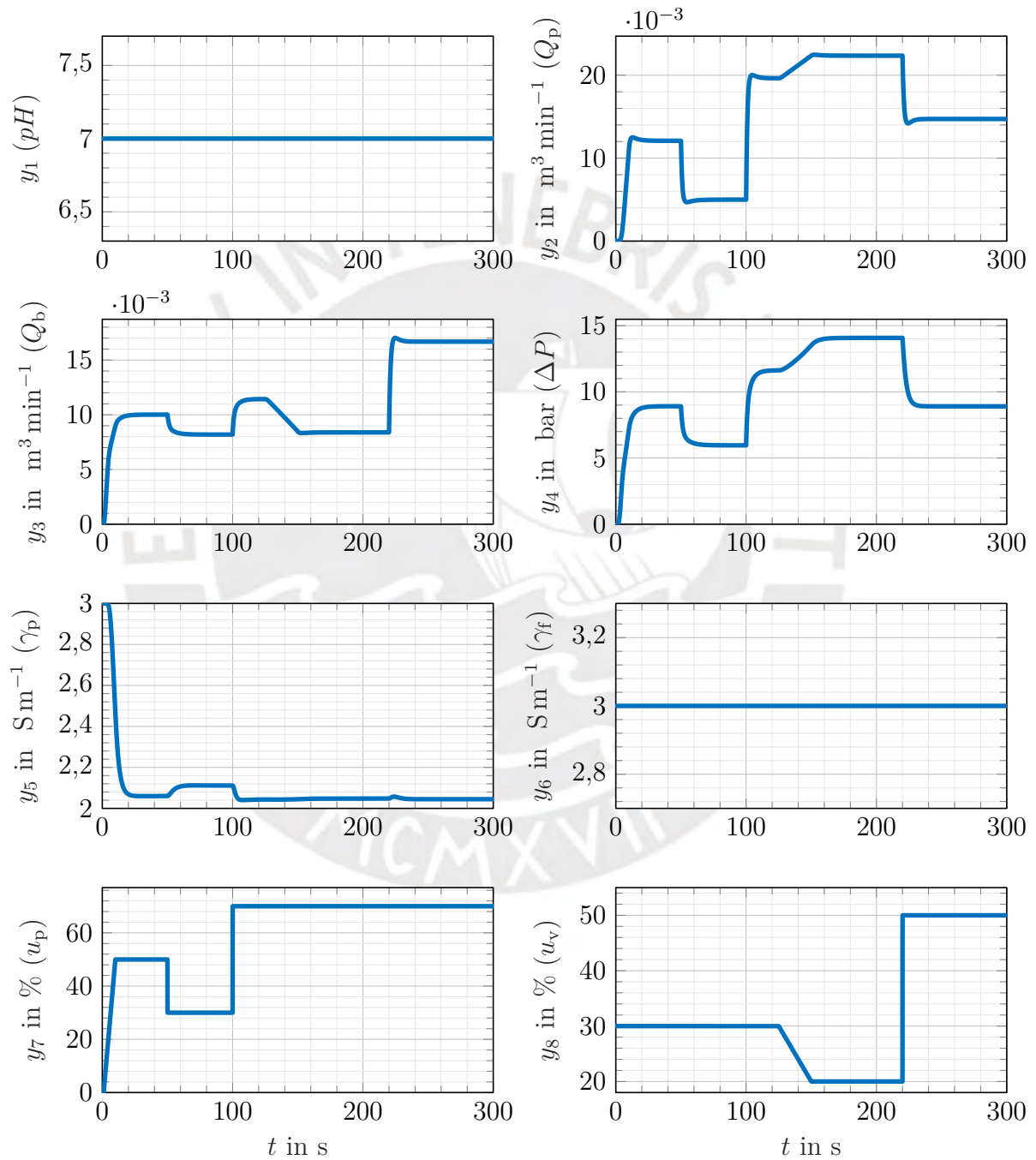


Fig. 3.5: Simulation of the RO system in open loop

Figure 3.5 shows the simulation of the RO system starting it from an initial point where the reverse osmosis has not taken place yet. This means that the rotational speed of the pump and the streams (see y_2 and y_3) need to be zero. Also the system pressure is zero (see y_4) and the salt concentrations on both sides on the membrane are equal that means $C_{ms} = C_b = C_p = C_f$. The measurements y_1 and y_6 are constant because they are actually no internal variables of the system (not affected by the inputs), but will be important parameters for fault diagnosis as they can vary over time.

y_1 is the pH value of the water of the feed stream and is dependent on the feed water source but also on the pre-treatment of the feed water. Note that the pH value is just required to determine the concentration of the respective stream because it affects its conductivity but not the actual value of the respective concentration.

However y_6 contains the pH value and the concentration C_f which is also dependent on the feed water source and affects also internal parameters like the brine concentration as shown before and the concentration on the membrane surface as seen in equation (3.37) and (3.42). If one can be sure that these two values are constant, their actual values can be replaced with the known values. Then the fault f_6 could also be seen as a feed water quality fault, as it affects just the value of y_6 but no other measured variables.

The actual reverse osmosis and its dynamics affected by the inputs of the system y_7 and y_8 can be seen with the measured outputs y_2 , y_3 , y_4 and y_5 . The first step is to rev up the motor of the pump, which causes the streams to flow and a pressure is generated. The stream y_2 has a steeper rise in the first seconds than y_3 but both reach their stationary value at about 20 seconds but y_2 does overshoot. Note that these two values are not actual states of the system as there is neither a \dot{Q}_p nor \dot{Q}_b . Their actual values depend on the given constraints of the system and the time dependence results from the states \dot{C}_p , \dot{C}_b , $\dot{\omega}_p$ and \dot{A}_v . An explanation for the fact that Q_b rises faster than Q_p can be seen in figure 3.4. When beginning with $\dot{\omega}_p = Q_f = 0$ and $C_{ms} = C_f$, the values Q_f and C_{ms} start on the left side of the red line where $Q_b > Q_p$. As they both rise with increasing Q_f and increasing ΔP , the ratio Q_b/Q_p decreases until $Q_b = Q_p$ which happens in this case at approximately 9 seconds. Afterwards, the permeate stream is always higher than the brine stream so the ratio Q_b/Q_p stays smaller than one.

The reason for the overshoot of y_2 cannot be seen directly with the measured pressure but with the behavior of the rotational speed ω_p or the feed stream Q_f (not shown here). In the simulation also Q_f overshoots because the motor of the pump accelerates faster than the concentrations have time to establish. As the brine stream only depends on the system pressure but the permeate stream on the system pressure and the membrane surface concentration, an overshoot occurs.

After 50 seconds the input u_p is set from 50 % to 30 %, which leads to a lower pressure and also the ratio Q_b/Q_p is greater than one again. The conductivity of the permeate stream y_5 does not seem highly affected by this change. However, it changes from its previous value of 2.06 S m^{-1} to 2.11 S m^{-1} which is equal to a concentration change from 0.6 kg m^{-3} to 1.12 kg m^{-3} .

Also the adjustment of u_p to 70 % changes the conductivity to 2.04 S m^{-1} which would be 0.43 kg m^{-3} .

Later the impact of changing the input u_v can be seen, where it is first set from 30 % to 20 %. When this happens, the ratio Q_b/Q_p decreases also and the system pressure rises, like when increasing u_p . The concentration of the permeate stream rises from 0.43 kg m^{-3} to 0.48 kg m^{-3} .

As one can see, a more closed brine valve leads to a smaller ratio of Q_b/Q_p . But it has the drawback that also the concentration of the permeate stream rises because of the higher osmotic pressure at the membrane, which results in less desalination of the feed stream. Nevertheless, the change is small since changing the valve opening to 50 %, the ratio Q_b/Q_p is again almost 1, the pressure decreases and the concentration of the permeate stream goes to 0.45 kg m^{-3} .

In the figures 3.6 - 3.8 the system is simulated with the faults $f_1 - f_3$ considered in section 3.6.2. The faults f_5 and f_6 are not simulated because they are only sensor faults and therefore do not change the system behavior in an open loop. The fault f_4 is not shown either because it has the same impact as changing the valve opening (suddenly or gradually) and can be seen in the simulation of the fault free plant in figure 3.5 from 120 s on. The two measurements y_1 and y_6 are not shown anymore because they are not changed during simulation and not affected by any considered fault.

Figure 3.6 shows the output behavior for a feed stream leakage of 11 min^{-1} . When the fault gets active, the pressure and both streams Q_b and Q_p decrease slightly and the permeate stream concentration rises. In figure 3.6 also the fault is shown so that one can see how the magnitude and the shape of the fault affects the system. But if it is not shown and this fault occurs, it is really difficult to detect it especially after the input change of u_p .

The impact of membrane fouling which is modeled with fault f_2 is shown in figure 3.7. The behavior of the system is totally different than with fault f_1 which is because the membrane fouling is not simulated with a step but with a ramp which changes the value of f_2 gradually over time. In this case the membrane surface gets one quarter clogged which means that 10 m^2 of the membrane surface are not usable anymore. In the last part from 250 s on, the two streams y_2 and y_3 do not converge anymore like before. They both diverge, while Q_p

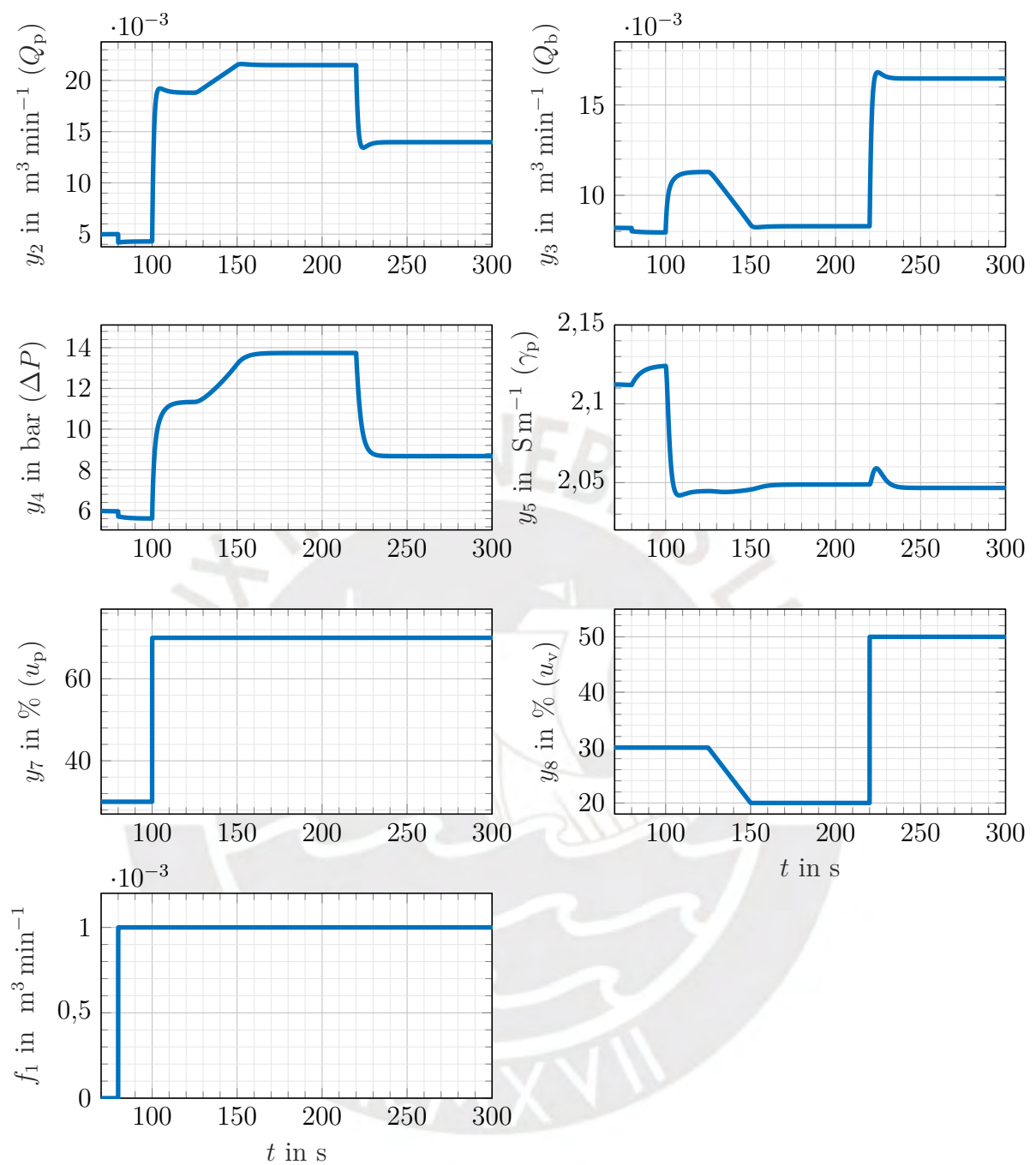
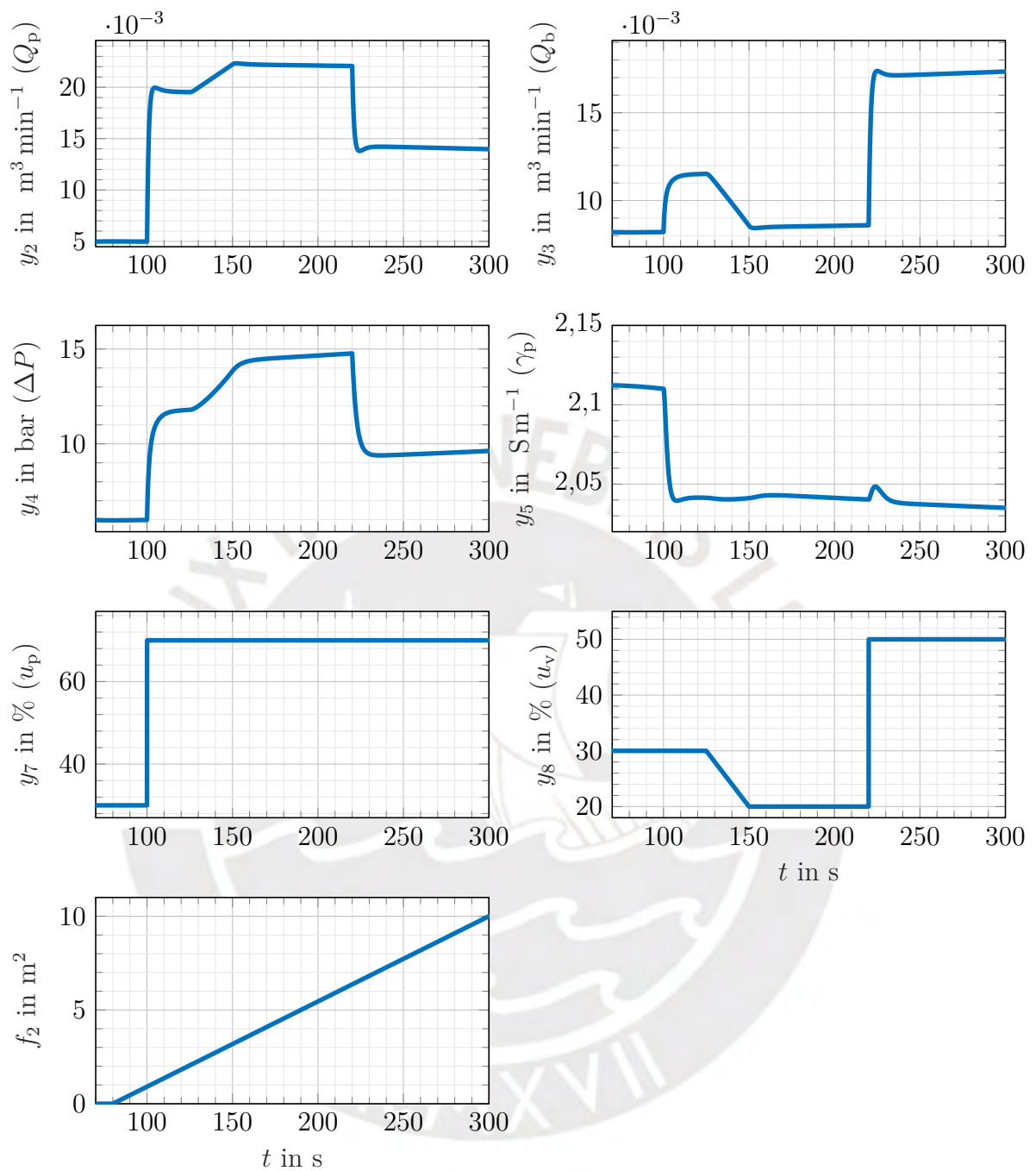
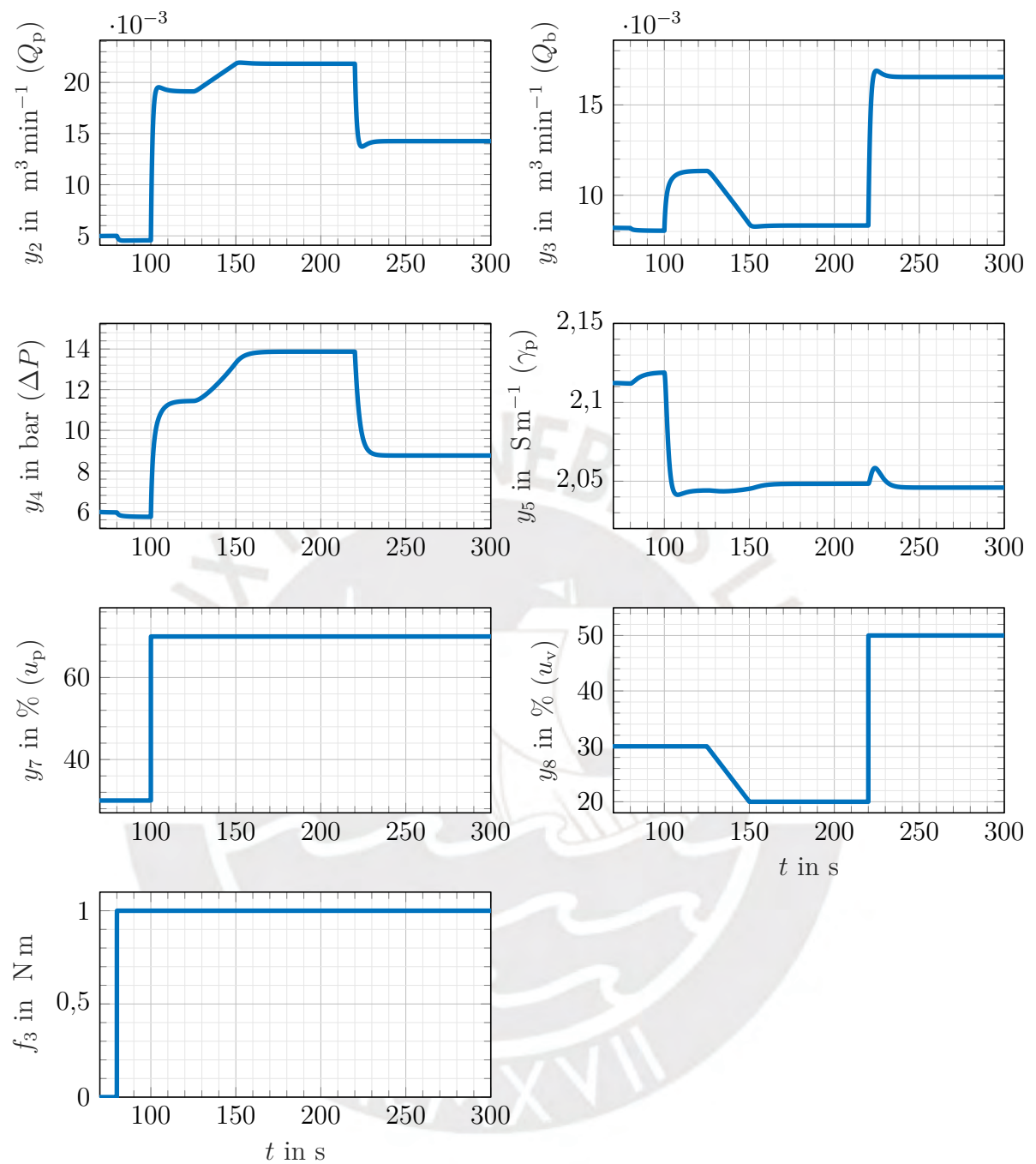


Fig. 3.6: Simulation of the RO system in open loop with fault f_1

decreases and Q_b increases which means the ratio Q_b/Q_p gets bigger. The pressure raises gradually and makes the salt concentration of the permeate stream decreasing. When fault f_2 reaches the 10 m^2 the concentration of the permeate stream is 0.35 kg m^{-3} .

The motor torque fault f_3 in figure 3.8 shows the same behavior as fault f_1 . Even though the fault can be detected, it will be difficult to distinguish it from fault f_1 .

Fig. 3.7: Simulation of the RO system in open loop with fault f_2

Fig. 3.8: Simulation of the RO system in open loop with fault f_3

The developed model shows reasonable simulation results. Unfortunately, it was not possible to compare the results with the real plant. Therefore, it is not possible to show exact values or sensor noise which will not be considered in this work. Additionally, the parameter values given in table 3.1 do not necessarily coincide with the real parameters of the plant and need to be determined before comparing with the model.

An advantage is that the new modeling approach by seeing the system pressure not as an independent controlled variable gives new possibilities not just for fault diagnosis but also for different controlling, estimation or optimization approaches.



Fault diagnosis via structural analysis

Structural analysis is an approach to determine the detectability/isolability of a fault for a given system which is modeled by a set of equations. With this method it is, for example, possible to assess if a given set of sensors is sufficient to detect and isolate all faults of interest.

The structural analysis made in this work is based on an algorithm used in [Pér17] for the fault diagnosis of distributed systems which can be used for centralized systems as well. The algorithm calculates the so called Fault-driven Minimal Structurally Overdetermined (FMSO) sets which can be used to deduce analytical redundancy relations (ARR). Based on the ARRs, online fault detection and isolation can be performed.

In the following chapter, at first an introduction to the mathematical background of the structural analysis is given. Afterwards the algorithm is presented and applied to the model developed in the previous chapter. This chapter finishes with the results of the structural analysis of the RO system.

4.1 Structural model

The model of a system can be seen in a wider sense as a set of constraints which describes the behavior of the system. The constraints couple the variables of the system in a mathematical way and can be of algebraic or differential type. Also other types like rules are possible, but in this work only the aforementioned is considered. The structural model used for structural analysis is another representation of the model and its constraints in form of a so called graph. In this graph, every constraint is only evaluated by the variables it contains, the way the variables are coupled in the sense of the mathematical operations is neglected. [EBP⁺19]

For example, the constraints

$$\begin{aligned} c_1 : x_1 &= e^{-x_1} - x_2 + f_1 \\ c_2 : x_2 &= y_1 \end{aligned} \quad (4.1)$$

contain the same variables as the constraints

$$\begin{aligned} c_1 : x_1 &= -3x_2 + f_1 \\ c_2 : x_2 &= 2y_1 \end{aligned} \quad (4.2)$$

Assume that y_1 is a known variable, f_1 a fault that can possibly occur and an explicit expression for x_1 in terms of y_1 and f_1 needs to be found. Both equation systems have a different expression for x_1 and in the first case calculating an explicit solution might not even be possible but both systems lead to the same structural model.

A special case are differential or integral equations. For example, in [FKÅ17] a system with a differential equation

$$\begin{aligned} c_1 : \dot{x}_1 &= -x_1 + 3x_2 + f_1 \\ c_2 : x_2 &= 2y_1 \end{aligned} \quad (4.3)$$

is split into two constraints with an extra variable into:

$$\begin{aligned} c_1 : x'_{11} &= -x_1 + 3x_2 + f_1 \\ c_2 : x_2 &= 2y_1 \\ c_3 : x'_{11} &= \dot{x}_1 \end{aligned} \quad (4.4)$$

The constraint c_3 in (4.4) is then considered particularly as a differential constraint which needs to be taken into account when selecting the set of equations for calculating the residuals. Another way considering differential constraints is given in [BKL⁺16] where the time derivative is just seen as an additional independent unknown variable.

The other approach is to neglect the differential constraint, where (4.3) leads to the same structure as the equation c_1 in (4.1) or (4.2). This approach is chosen in this work and can lead to some difficulties which will be discussed later on in section 4.5.1 when the ARRs are presented.

Seeing the constraints and variables of an equation system as elements of a set and the connections between the elements as another set, the structural representation of the system can be made as a graph which is defined as follows:

Definition 4.1.1. (Finite graph) [Pot19]

A finite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \varphi)$ is a pair of sets consisting of a nonempty finite set \mathcal{V} (called **vertices**) and a finite set \mathcal{E} (called **edges**) and a mapping

$$\varphi : \mathcal{E} \longrightarrow \{\mathcal{N} \subseteq \mathcal{V} \mid |\mathcal{N}| \in \{1, 2\}\}$$

which assigns (at most) two vertices to every edge in \mathcal{E} . Two connected vertices are also said to be **adjacent**.

The graph of the structural model of (4.1) and (4.2) can then be written as:

$$\mathcal{G} = (\{c_1, c_2, x_1, x_2, y_1, f_1\}, \{1, 2, 3, 4, 5\}, \varphi) \quad (4.5)$$

with:

$$\varphi = \{(1, \{c_1, x_1\}), (2, \{c_1, x_2\}), (3, \{c_1, f_1\}), (4, \{c_2, x_2\}), (5, \{c_2, y_1\})\}$$

The resulting graph has some characteristics which are important for the structural analysis. Firstly, no vertex is connected to itself which would be for example $\varphi = \{(1, \{c_1\}), \dots\}$. Therefore, it holds that $|\mathcal{N}| = 2$ where $|\mathcal{N}|$ is called the cardinality of N which is the number of elements in N .

Furthermore, no pair of vertices is connected more than once which would be for example $\varphi = \{(1, \{c_1, x_1\}), (2, \{c_1, x_1\}), \dots\}$. Thus, it holds that the mapping φ is injective [Pot19].

A graph with the aforementioned characteristics is also called a simple graph, but the graphs of a structural model have another characteristic which is important for the structural analysis for fault diagnosis. The set of vertices \mathcal{V} can be divided into two disjoint subsets,

$$\mathcal{V} = \mathcal{V}_C \cup \mathcal{V}_V \quad (4.6)$$

with \mathcal{V}_C the set of constraints and \mathcal{V}_V the set of variables and within these two sets none of the vertices are adjacent to each other.

Therefore it holds that $\mathcal{V}_C \cap \mathcal{V}_V = \emptyset$ and $\varphi : \mathcal{K} \rightarrow \{ \{v, c\} \mid v \in \mathcal{V}_V, c \in \mathcal{V}_C \}$. [Lev19]

A graph with the three aforementioned characteristics is called a bipartite graph which is defined for a structural model as follows:

Definition 4.1.2. (Bipartite graph) [EBP⁺19]

Let \mathcal{V}_C be the set of constraints of a model and \mathcal{V}_V the set of variables of the model. Then the structural model can be represented by the bipartite graph $\mathcal{G} = (\mathcal{V}_C \cup \mathcal{V}_V, \mathcal{E})$. Where \mathcal{E} is a set of edges between vertices in the two sets of vertices \mathcal{V}_C and \mathcal{V}_V . An edge $\{c_j, v_j\} \in \mathcal{E}$ exists if variable $v_j \in \mathcal{V}_V$ appears in the constraint $c_j \in \mathcal{V}_C$.

With this notation, the three sets of the graph for system (4.1) and (4.2) are $\mathcal{V}_C = \{c_1, c_2\}$, $\mathcal{V}_V = \{x_1, x_2, y_1, f_1\}$ and $\mathcal{E} = \{\{c_1, x_1\}, \{c_1, x_2\}, \{c_1, f_1\}, \{c_2, x_2\}, \{c_2, y_1\}\}$. Note that this simplified notation can also be used to represent a graph like defined in 4.1.1. But then \mathcal{E} is not necessarily a set but a collection because elements can appear more than once in \mathcal{E} [And02]. Graphs can also be illustrated as shown for this bipartite graph in figure 4.1. The vertices, also called nodes, in \mathcal{V}_V are drawn as circles and the vertices in \mathcal{V}_C as rectangles. An edge is the connection between two vertices. The illustration of this graph is not unique, as for example the vertices and edges could be placed differently. In this case it does not matter how they are placed. However, there are fields in graph theory, for example graph coloring, where it matters. [Lev19]

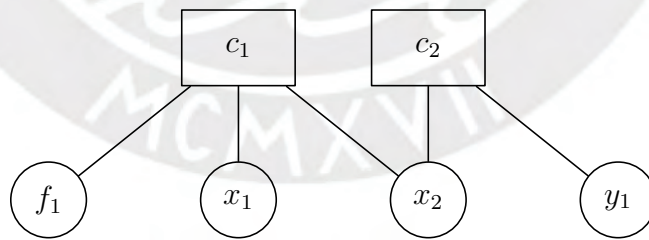


Fig. 4.1: Bipartite graph of system (4.1) and (4.2)

Another way to visualize a structural model is the so-called biadjacency matrix. In a biadjacency matrix the number of rows is determined by the number of constraints and the number of columns is the number of elements in \mathcal{V}_V . The elements of the matrix are then equal to the edges of the bipartite graph. If the variable $v_j \in \mathcal{V}_V$ appears in the constraint $c_i \in \mathcal{V}_C$, the element is non-empty (in this work marked with X). If it does not appear, the

element is empty [EBP⁺19]. The biadjacency matrix to the graph in figure 4.1 is given in equation (4.7).

$$\begin{array}{c|c|c|c|c}
 & x_1 & x_2 & f_1 & y_1 \\
 \hline
 c_1 & X & X & X & \\
 \hline
 c_2 & & X & & X
 \end{array} \tag{4.7}$$

In equation 4.7 the variables are divided into three groups: the known and unknown variables and the faults. Also the set \mathcal{V}_V can be partitioned into

$$\mathcal{V}_V = \mathcal{V}_{VY} \cup \mathcal{V}_{VX} \cup \mathcal{V}_{VF} \tag{4.8}$$

where \mathcal{V}_{VY} is the set of known, \mathcal{V}_{VX} the set of unknown variables and \mathcal{V}_{VF} the set of faults. In this case, is $\mathcal{V}_Y = \{y_1\}$, $\mathcal{V}_X = \{x_1, x_2\}$ and $\mathcal{V}_F = \{f_1\}$. In the same way the constraints can be partitioned into the subsets \mathcal{V}_{CY} and \mathcal{V}_{CX} so that holds:

$$\mathcal{V}_C = \mathcal{V}_{CY} \cup \mathcal{V}_{CX} \tag{4.9}$$

Where \mathcal{V}_{CY} includes all constraints that just contain known variables and \mathcal{V}_{CX} all constraints that contain at least one or more unknown variables. Note that a constraint with only faults is not considered. For the structural model of equation (4.7) it holds that $\mathcal{V}_{CY} = \emptyset$ because there is no constraint with just known variables. A constraint where all variables are known, results for example from a control law where the measured outputs are used to calculate the new input of the system. [BKL⁺16]

Based on the partition of \mathcal{V}_C and \mathcal{V}_V , a graph $\mathcal{G} = (\mathcal{V}_C \cup \mathcal{V}_V, \mathcal{E})$ can be split into the three subgraphs $\mathcal{G}_X = (\mathcal{V}_{VX} \cup \mathcal{V}_{CX}, \mathcal{E}_X)$, $\mathcal{G}_F = (\mathcal{V}_{VF} \cup \mathcal{V}_{CX}, \mathcal{E}_F)$ and $\mathcal{G}_Y = (\mathcal{V}_{VY} \cup \mathcal{V}_{CY}, \mathcal{E}_Y)$. A subgraph is defined as follows:

Definition 4.1.3. (Subgraph) [Lev19]

$\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ is a **subgraph** of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ if $\mathcal{V}' \subseteq \mathcal{V}$ and $\mathcal{E}' \subseteq \mathcal{E}$. A subgraph can be written as $\mathcal{G}' \subseteq \mathcal{G}$.

An **induced subgraph** is a subgraph where every edge in \mathcal{E} is also in \mathcal{E}' if the two adjacent vertices are also in \mathcal{V}' .

The subgraphs taken for structural analysis are always induced subgraphs, as it is necessary for finding the right links between the unknown variables and the given constraints. It is not useful, to eliminate an edge in the subgraph as it is comparable to eliminating the appearance of a variable in a constraint. The resulting subgraphs of the structural model are again bipartite.

For the structural analysis only the subgraph \mathcal{G}_X with the unknown variables is used, as it is the aim to find a set of constraints that makes it possible to eliminate all the unknown variables [BKL⁺16]. Therefore, in the following the set \mathcal{V}_C refers to the set of constraints with at least one unknown variable and \mathcal{V}_X the set of unknown variables of the system. An important tool of the structural analysis are matchings which are introduced in the next section.

4.2 Matching

A typical example for matchings is the so called marriage problem. The sets of vertices of the bipartite graph comprises the set of men and the set of women that want to marry. All men and women are now asked to make a list with the ones they would marry. The list can contain multiple names or just one, but every name needs to be of the opposite gender. There will be an edge between two vertices if both, man and woman, agree to a marriage. Thus, every vertex can be adjacent to multiple vertices from the other set.

A matching then shows the different possible marriages if polygamy is not an option. This means that every vertex can be matched to only one vertex from the other set, but it does not imply that every vertex needs to be matched, which would be if all men and women could marry. [Lev19]

Definition 4.2.1. (Matching) [BKL⁺16]

A matching $\mathcal{M} \subseteq \mathcal{E}$ is a set of disjoint edges of a bipartite graph. Where disjoint means that all the edges have no vertex in common.

Figure 4.2 shows different matchings of the graph shown in figure 4.1. The edges of the matching are here drawn as bold lines. There exist matchings with just one edge, for example $\mathcal{M}_1 = \{c_1, x_1\}$, but also matchings with two edges, for example $\mathcal{M}_6 = \{e_1, x_1\}, \{e_2, x_2\}$. The matchings \mathcal{M}_5 and \mathcal{M}_6 are also called a **complete matching** which is defined as follows:

Definition 4.2.2. (Complete matching) [BKL⁺16]

A matching \mathcal{M} of a bipartite graph $\mathcal{G} = (\mathcal{V}_C \cup \mathcal{V}_X, \mathcal{E})$ is complete with respect to \mathcal{V}_C if holds $|\mathcal{M}| = |\mathcal{V}_C|$ or complete with respect to \mathcal{V}_X if $|\mathcal{M}| = |\mathcal{V}_X|$.

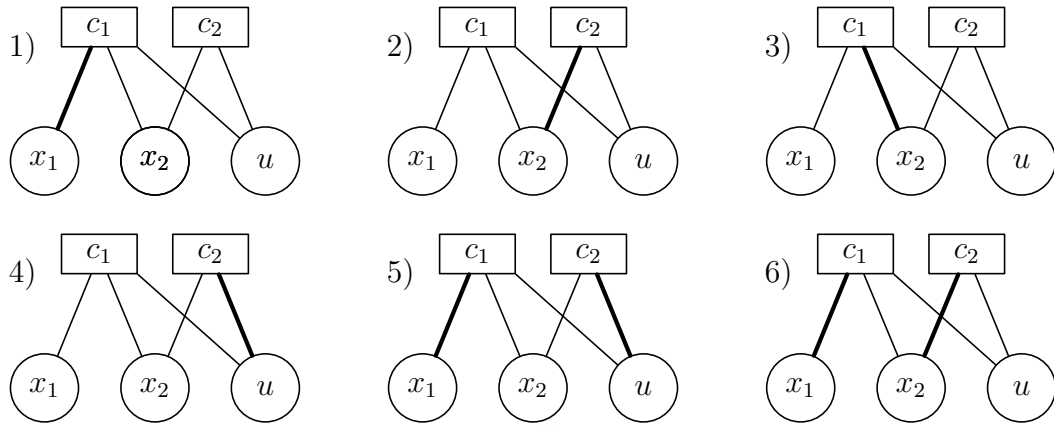


Fig. 4.2: Different matchings for the bipartite graph in 4.1

Basically a matching is complete if no more edges of \mathcal{E} can be added to \mathcal{M} without violating the definition in 4.2.1. For example, adding the edge $\{c_2, u\}$ to \mathcal{M}_2 is a violation because the vertex c_2 then belongs to two edges in the matching.

For the structural analysis finding a complete matching of \mathcal{V}_C (or induced subgraphs of the system with vertices in \mathcal{V}_C) is more of interest because a matching equals connecting a variable to a constraint. A system usually has equal or more constraints than unknown variables. It is assumed that the connected constraint makes it possible to calculate the variable if all the other variables of the constraint are known. For example, in matching \mathcal{M}_5 constrained c_2 is used to determine x_2 . If u is a known variable it is possible to calculate it with just this constraint. x_1 can be calculated with c_1 , but x_2 is required. x_2 is known through c_2 and so on.

Whether a bipartite graph has a complete matching or not, can be proven with the following theorem from Philip Hall [Hal35]:

Theorem 4.2.1. (Hall's Marriage Theorem) [Lev19]

Let \mathcal{G} be a bipartite graph with the sets \mathcal{V}_X and \mathcal{V}_C . Then \mathcal{G} has a complete matching with respect to \mathcal{V}_X if and only if

$$|N(S)| \geq |S|$$

for all $S \subseteq \mathcal{V}_X$. Where $N(S)$ is the set of vertices connected to the vertices in S .

The condition $|N(S)| \geq |S|$ is also called the **marriage condition**. In order to understand this theorem, three different graphs are shown in figure 4.3. The theorem states that there exists a complete matching if and only if for the graph itself and all induced subgraphs the condition $|N(S)| \geq |S|$ holds.

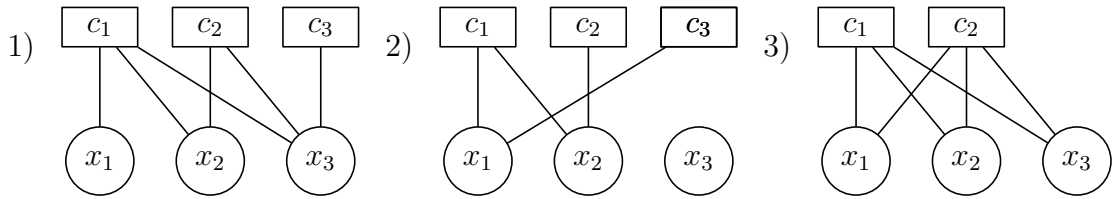


Fig. 4.3: Example graphs for the Hall's Marriage Theorem

In the following is analyzed if a complete matching exists for the graphs $\mathcal{G}_1 - \mathcal{G}_3$ and if there is a complete matching, with respect to which set.

For the graph \mathcal{G}_1 taking first $S = \mathcal{V}_X = \{x_1, x_2, x_3\}$, which is the complete graph, leads to $N(S) = \mathcal{V}_C = \{c_1, c_2, c_3\}$ as the tree vertices in S are connected to every edge in \mathcal{V}_C . Thus $|S| = 3$ and $|N(S)| = 3$, which means that the condition $|N(S)| \geq |S|$ holds. Taking $S = \{x_1, x_3\}$ or $S = \{x_2, x_3\}$ also leads to $N(S) = \{c_1, c_2, c_3\}$, thus $|N(S)| = 3 > 2 = |S|$. $S = \{x_1, x_2\}$ leads to $|S| = 2$ and $|N(S)| = 2$ which also fulfills the condition in theorem 4.2.1. Furthermore, every single vertex in \mathcal{V}_X needs to have at least one connection to a vertex in \mathcal{V}_C which is the case for the graph \mathcal{G}_1 . Therefore, the graph has a complete matching in \mathcal{V}_X and also in \mathcal{V}_C as the condition also holds for all $S \subseteq \mathcal{V}_C$.

That this is not always the case shows graph \mathcal{G}_2 where the marriage condition holds for every subset $S \subseteq \mathcal{V}_X$ except for $S = \{x_3\}$, there are $|S| = 1$ and $|N(S)| = 0$. \mathcal{G}_2 has neither in \mathcal{V}_X nor in \mathcal{V}_C a complete matching. This case is excluded in a structural model because every variable is connected to a constraint and there is no constraint without a variable. This fact is assumed for the further definitions.

In graph \mathcal{G}_3 , for $S = \mathcal{V}_X$ the condition fails. There is only a complete matching for \mathcal{V}_C where even holds $|S| < |N(S)|$.

A complete graph of a structural model might not necessarily have a complete matching in \mathcal{V}_X or \mathcal{V}_C , but that might not be necessary for the aim of fault diagnosis. Often it is sufficient if the induced subgraphs of the model contain a complete matching. Therefore, the so called Dulmage-Mendelsohn composition is introduced in the next section.

4.3 Dulmage-Mendelsohn decomposition

Based on Hall's marriage theorem, the Dulmage-Mendelsohn (DM) decomposition characterizes, the bipartite graph according to its structural property. In [DM58] it is shown, that a bipartite graph can be divided into three subgraphs with the following properties:

Theorem 4.3.1. (Dulmage-Mendelsohn decomposition of bipartite graphs) [BKL⁺16]

Each bipartite graph $\mathcal{G} = (\mathcal{V}_C \cup \mathcal{V}_X, \mathcal{E})$ can be decomposed into three subgraphs with the following properties:

- **Over-constrained subgraph \mathcal{G}^+** , which has a complete matching in \mathcal{V}_X but not in \mathcal{V}_C .
- **Just-constrained subgraph \mathcal{G}^0** , which has a complete matching in \mathcal{V}_C and \mathcal{V}_X .
- **Under-constrained subgraph \mathcal{G}^-** , which has a complete matching in \mathcal{V}_C but not in \mathcal{V}_X .

Based on this partition also the two sets of vertices \mathcal{V}_X and \mathcal{V}_C and the set of edges are partitioned so that holds:

$$\mathcal{V}_C = \mathcal{V}_C^- \cup \mathcal{V}_C^0 \cup \mathcal{V}_C^+ \quad (4.10)$$

$$\mathcal{V}_X = \mathcal{V}_X^- \cup \mathcal{V}_X^0 \cup \mathcal{V}_X^+ \quad (4.11)$$

$$\mathcal{E} = \mathcal{E}^- \cup \mathcal{E}^0 \cup \mathcal{E}^+ \quad (4.12)$$

Figure 4.4 shows an example of the Dulmage-Mendelsohn decomposition. The bipartite

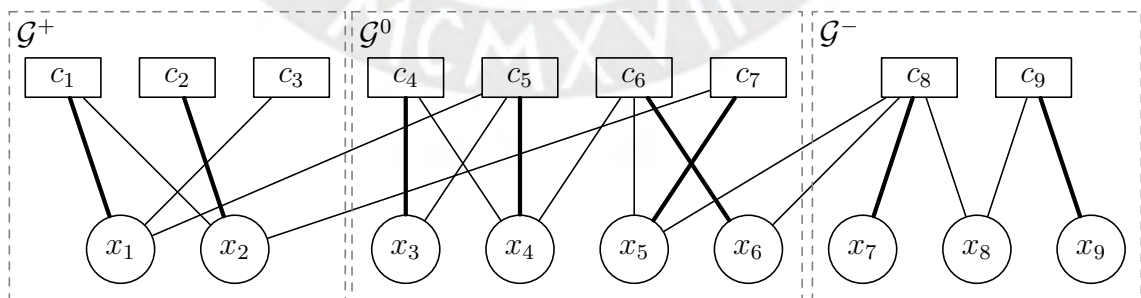


Fig. 4.4: Example for the Dulmage-Mendelsohn decomposition [BKL⁺16]

graph can be partitioned into the subgraphs \mathcal{G}^- , \mathcal{G}^0 and \mathcal{G}^+ with the respective sets of vertices and edges shown in equation (4.13) and (4.14). The matchings, which are also

shown in figure 4.4, are here marked as a circle around the edge which is a way to illustrate matchings in a bi-adjacency matrix. Unlike matchings the DM decomposition is always unique with respect to the vertices of the subgraphs but it is not certain that it holds for all the subgraphs that their set $\mathcal{V} \neq \emptyset$. [BKL⁺16]

$$\mathcal{G}^+ = \begin{array}{c|cc} & x_1 & x_2 \\ \hline c_3 & X & \\ c_1 & \textcircled{X} & X \\ c_2 & & \textcircled{X} \end{array} \quad \mathcal{G}^- = \begin{array}{c|ccc} & x_7 & x_9 & x_8 \\ \hline c_8 & \textcircled{X} & & X \\ c_9 & & \textcircled{X} & X \end{array} \quad (4.13)$$

$$\mathcal{G}^0 = \begin{array}{c|cccc} & x_3 & x_4 & x_5 & x_6 \\ \hline c_4 & \textcircled{X} & X & & \\ c_5 & X & \textcircled{X} & & \\ c_7 & & & \textcircled{X} & \\ c_6 & & X & X & \textcircled{X} \end{array} \quad (4.14)$$

In the matrices in (4.13) the constraints and variables are not in an ascending order which leads to a diagonal arrangement of the matchings of the subgraph. Switching the lines and columns of the whole graph \mathcal{G} according to the two subgraphs leads to the illustration shown in equation (4.15).

The **blue** fields mark the subgraphs \mathcal{G}^+ , \mathcal{G}^0 and \mathcal{G}^- which are also on a diagonal of the matrix. This diagonal partitions the graph into two fields, here shown in red and green, which have certain characteristics shown by the DM decomposition.

The **red** part does not contain any edges of the graph whereas the **green** part has edges which do not appear in the subgraphs. With this knowledge it is possible to determine the subgraphs \mathcal{G}^+ , \mathcal{G}^0 and \mathcal{G}^- of the DM decomposition by just swaping the rows and columns of the incidence matrix.

Another characteristic of the DM decomposition is its maximal matching which can be determined easily by marking the edges on the diagonal beside the main diagonal. A maximal matching \mathcal{M}_{\max} is a matching which contains the maximal number of edges of a graph without violating the definition of a matching. Considering the marriage problem introduced in section 4.2, a maximal matching indicates the maximal number of men (or women) that can marry, which is equal to $|\mathcal{M}_{\max}|$. Like all matchings also a maximal matching does not need to be unique as in this example. Matching the edge $\{c_9, x_8\}$ instead of $\{c_9, x_9\}$ still fulfills the definition of a matching.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_9	x_8
c_3	X								
c_1	X	X							
c_2		X							
c_4			X	X					
c_5	X		X	X					
c_7		X			X				
c_6				X	X	X			
c_8					X	X	X		X
c_9								X	X

(4.15)

The subgraph \mathcal{G}^+ is interesting for fault diagnosis, because it has more equations than variables. Based on the system of equations which resulted in the subgraph \mathcal{G}^+ , an over-determined system of equations has more equations than variables. This also means that the equation system has either **one** or **none** solution which is used to check the consistency of the system. [Pér17]

For example, the system

$$\begin{aligned}
 c_1 : 0 &= x_1 - 4x_2 \\
 c_2 : 0 &= x_2 + y_2 \\
 c_3 : 0 &= x_1 - y_1
 \end{aligned}
 \tag{4.16}$$

where y_1 and y_2 are variables and x_1, x_2 are unknown variables. Not taking into account y_1 and y_2 in the structural model leads to a graph like \mathcal{G}^+ of the example before. The equation system can now be used to check the consistency of the system with the model.

The system is consistent with the model if the measured variables are a solution of the equation system. For example, this is true if the measured variables are $y_1 = 1$ and $y_2 = -0.25$. Then the constraint c_1 leads to $1 - 4 \cdot 0.25 = 0$ thus the measured variables are consistent with the system. If the next values of the measurement are $y_1 = 1$ and $y_2 = 2$, which results in constraint c_1 to $1 - 4 \cdot (-2) = 9 \neq 0$, this means that the system is not consistent with the model and thus it needs to be assumed that the system is faulty.

In this case it is simple to calculate a relation which checks the consistency by substituting x_1 and x_2 obtained from c_2 and c_3 in c_1 which leads to $0 = y_1 + 4y_2$. If this relation (later called ARR) is violated, the system is not consistent with the model.

Therefore, finding over-constrained subgraphs of a structural model in order to calculate residuals is the main aim of the structural analysis for fault diagnosis. Obviously, one could try to randomly rearrange the rows and columns of the bi-adjacency matrix until the structure of the DM decomposition is achieved. However, nowadays there exist more

efficient algorithms which one of them will be presented in section 4.4.1. Note that the graph \mathcal{G}^+ has a complete matching in \mathcal{V}_X **but not** in \mathcal{V}_C . But a graph with a complete matching in \mathcal{V}_X **but not** in \mathcal{V}_C is not necessarily an over-constrained graph because the graph $\mathcal{G}^+ \cup \mathcal{G}^0$ also has this characteristics.

Before the actual algorithm, used in this work, is presented, some definitions used in most of the works which treat the fault diagnosis via structural analysis need to be given to avoid misunderstandings.

Until now a system model in form of equations can be represented as a bipartite graph with the constraints and variables as disjoint sets of vertices and an edge of the graph exists if a variable appears in the constraint. As the main aim is finding an over-constrained subgraph of the structural model, which also means that there are more constraints than variables, a set of equations can be characterized as follows:

Definition 4.3.1. (Structurally Overdetermined set) [KÅN05]

Given a set of constraints $S \subseteq \mathcal{V}_C$ which contains a set of unknown variables $N(S) \subseteq \mathcal{V}_X$. The set S is called structurally overdetermined (SO) if $|S| > |N(S)|$.

Note that a structurally overdetermined set is not automatically the set of vertices of an over-constrained graph as shown in the following example.

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
$c_1 : 0 = x_1 - y_1$	c_5			X				
$c_2 : 0 = x_1 - 4x_2$	c_1	X						
$c_3 : 0 = x_2 + y_2$	c_2	⊗	X					
$c_4 : 0 = x_2 + x_3 - x_4$	c_3		⊗					
$c_5 : 0 = x_3 + y_4$	c_4		X	⊗	X			
$c_6 : 0 = x_4 + y_5$	c_6				⊗			
$c_7 : 0 = x_5 + x_7$	c_7					⊗		X
$c_8 : 0 = x_6 + x_7$	c_8						⊗	X

(4.17)

In (4.17) on the left side an equation system is shown and on the right side its corresponding biadjacency matrix with the structure of the DM decomposition. It can be seen immediately that the set of equations is structurally overdetermined as there is one more constraint than variables. However, the graph is not over-constrained because there is no complete matching in $S = \mathcal{V}_X$.

Therefore, the following definition is given:

Definition 4.3.2. (Proper Structurally Overdetermined set) [KÅN05]

A structurally overdetermined set of constraints $S \subseteq \mathcal{V}_C$ with the set of unknown variables $N(S) \subseteq \mathcal{V}_X$ is a proper structurally overdetermined (PSO) set if $S = S^+$. Where the structurally overdetermined set $S^+ \subseteq \mathcal{V}_C$ are the vertices of an over-constrained graph $\mathcal{G}^+(S^+ \cup N(S^+), \mathcal{E})$. Thus, for any maximum matching of \mathcal{G}^+ every constraint $s \in S^+$ can be reached by an **alternating path** between one unmatched constraint and s .

To understand this definition, the definition of an alternating path is required, which is taken from [PF90] where an algorithm for finding the DM decomposition is explained. The actual algorithm will be shown in section 4.4.1 with more details.

It says that [PF90]:

Taking a bipartite Graph $\mathcal{G} = (\mathcal{V}_C \cup \mathcal{V}_X, \mathcal{E})$ and $v \in \mathcal{V}$, where $\mathcal{V} = \mathcal{V}_C \cup \mathcal{V}_X$.

A **walk** is a sequence of vertices $v_0, v_1, \dots, v_{n-1}, v_n$ such that $\{v_i, v_{i+1}\}$ is an edge of \mathcal{G} , thus $\{v_i, v_{i+1}\} \in \mathcal{E}$. Vertices or edges can be repeated in a walk. Considering that it is a bipartite graph it holds that after a vertex from \mathcal{V}_C comes a vertex from \mathcal{V}_X and vice versa. An **alternating walk** is a walk with alternate edges in a matching \mathcal{M} . That means that if $\{v_i, v_{i+1}\} \in \mathcal{M}$, $\{v_{i+1}, v_{i+2}\} \in \{\mathcal{E}/\mathcal{M}\}$, $\{v_{i+2}, v_{i+3}\} \in \mathcal{M}$, $\{v_{i+3}, v_{i+4}\} \in \{\mathcal{E}/\mathcal{M}\}$ and so on.

An **alternating tour** is an alternating walk whose start and end vertex is the same.

An **alternating path** is an alternating walk without repeated vertices.

Now the following example is taken:

$$\mathcal{G}^+ = \begin{array}{c|cc} & x_1 & x_2 \\ \hline c_1 & X & \\ \hline c_2 & X & X \\ \hline c_3 & & X \\ \hline c_4 & & X \end{array} \quad (4.18)$$

It can be seen immediately that the graph in (4.18) has a complete matching in \mathcal{V}_X but not in \mathcal{V}_C . This also holds for the subgraphs with the set of constraints $\mathcal{V}_C = \{c_1, c_2, c_3\}$, $\{c_2, c_3, c_4\}$ and $\{c_3, c_4\}$. In order to show what is an alternating path another visual representation of bipartite graphs is shown in figure 4.5. Figure 4.5 shows the graph of equation (4.18) and all its subgraphs which are over-constrained as a tree structure. The tree structure is not necessary here, but it is convenient for the visualization of the graph and its

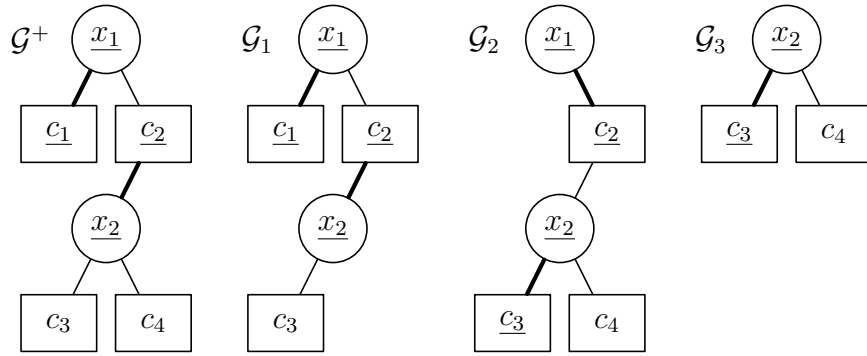


Fig. 4.5: Bipartite graph and over-constrained subgraphs of equation (4.18)

paths. For all the graphs a maximal matching is shown, where the matched edges are bold and the matched vertices are underlined. Now the alternating paths can be explained as follows:

Starting with the complete graph \mathcal{G}^+ which has two unmatched vertices c_3 and c_4 . An alternating path starting with an edge that is not in a maximal matching \mathcal{M} , means that the second edge needs to be an element of \mathcal{M} . Therefore, starting with the unmatched vertex c_3 leads at least to the path c_3, x_2, c_2 . The alternating path can be continued as there is an edge in \mathcal{E}/\mathcal{M} connected to c_2 . Starting with c_3 and going until the end leads to the alternating path c_3, x_2, c_2, x_1, c_1 . The path for c_4 is c_4, x_2, c_2, x_1, c_1 , thus all vertices of \mathcal{V}_C appearing in these two paths are the PSO set, which is $S^+ = \{c_1, c_2, c_3, c_4\}$.

As in \mathcal{G}_1 the maximal alternating path is c_3, x_2, c_2, x_1, c_1 , the PSO set is $S_1^+ = \{c_1, c_2, c_3\}$. For the graph \mathcal{G}_2 , starting with the unmatched vertex c_4 leads to the path c_4, x_2, c_3 , thus its set of vertices \mathcal{V}_C is not a PSO set, even though the graph \mathcal{G}_2 has a complete matching in \mathcal{V}_X but not in \mathcal{V}_C . The PSO set includes only $\{c_3, c_4\}$ which is the same as in graph \mathcal{G}^3 , therefore $S_2^+ = S_3^+ = \{c_3, c_4\}$. This occurs because the graph \mathcal{G}^2 can be further decomposed into a structurally over-constrained graph and a just constraint graph with the vertices $\mathcal{V}_X = \{x_1\}$ and $\mathcal{V}_C = \{c_2\}$.

It can be seen that S_1^+ , S_2^+ and S_3^+ are all subsets of S^+ , but all are PSO sets. Therefore a proper structurally overdetermined set of constraints can be still further partitioned until a minimal structurally overdetermined set is obtained which is defined as follows:

Definition 4.3.3. (Minimal Structurally Overdetermined set) [KÅN05]

A proper structurally overdetermined set of constraints is a minimal structurally overdetermined (MSO) set if no proper subset is structurally overdetermined.

From the definition 4.3.3 it follows that for the graph \mathcal{G}^+ of equation (4.18) just the sets $c_1, c_2, c_3, c_1, c_2, c_4$ and c_3, c_4 are MSO sets. Looking at the subgraphs of the MSO sets, it can be seen that there is always one variable less than constraints in the subgraph whereas in the graph \mathcal{G}^+ there are two more variables than constraints. Therefore, the structural redundancy is introduced.

Definition 4.3.4. (Structural redundancy) Given a bipartite graph $\mathcal{G} = (\mathcal{V}_C \cup \mathcal{V}_X, \mathcal{E})$ and the sets $S \subseteq \mathcal{V}_C$ and $N(S) \subseteq \mathcal{V}_X$ where $N(S)$ is the set of vertices adjacent to at least one vertex in S . Then the structural redundancy is defined as:

$$\rho_S = |S| - |N(S)|$$

In the marriage theorem 4.2.1 the value of ρ_S was used to determine if there is a complete matching with respect to \mathcal{V}_X or \mathcal{V}_C . Restricting S to be $S \subseteq \mathcal{V}_C$, it can be checked if S is a MSO set via the structural redundancy, because the structural redundancy of a MSO set is 1 and for PSO sets it can be greater than 1. [Kry06]

Until now faults have not been considered in the structural analysis of the system, because they are not taken into account when looking for MSOs.

This is where the idea of fault-driven minimal structurally overdetermined (FMSO) is considered. A FMSO is defined as:

Definition 4.3.5. (Fault-driven Minimal Structurally Overdetermined set) [PTC⁺15]

Let \mathcal{V}_C be a set of constraints and \mathcal{F} all the faults appearing within the constraints \mathcal{V}_C . Then a minimal structurally overdetermined set is a fault-driven minimal structurally overdetermined (FMSO) set $S \subseteq \mathcal{V}_C$. If the set $\mathcal{F}_S \subseteq \mathcal{F}$ which are the faults appearing in the constraints S is not empty, hence $\mathcal{F}_S \neq \emptyset$.

The definition of an FMSO is explained with the following example:

$$\begin{array}{l}
 c_1 : 0 = x_1 - y_1 \\
 c_2 : 0 = x_1 - 4x_2 + f_2 \\
 c_3 : 0 = x_2 + y_2 \\
 c_4 : 0 = x_2 + y_3 + f_1 \\
 c_5 : 0 = x_2 + x_3 - x_4 \\
 c_6 : 0 = x_3 + y_4 \\
 c_7 : 0 = x_4 + y_5
 \end{array}
 \quad
 \begin{array}{c|cccc|cc}
 & x_1 & x_2 & x_3 & x_4 & f_1 & f_2 \\
 \hline
 c_1 & X & & & & & \\
 \hline
 c_2 & X & X & & & & X \\
 \hline
 c_3 & & X & & & & \\
 \hline
 c_4 & & X & & & X & \\
 \hline
 c_5 & & X & X & X & & \\
 \hline
 c_6 & & & X & & & \\
 \hline
 c_7 & & & & X & & \\
 \hline
 \end{array}
 \tag{4.19}$$

At first the MSOs are searched in the left part (column $x_1 - x_4$) of the matrix in (4.19). Known variables are not shown in the matrix because they are not considered for finding (F)MSOs. The system of equations contains 6 MSOs and the following four will be further evaluated $M_1 = \{c_3, c_4\}$, $M_2 = \{c_1, c_2, c_3\}$, $M_3 = \{c_1, c_2, c_4\}$ and $M_4 = \{c_3, c_5, c_6, c_7\}$. Starting with M_1 which contains constraint c_4 where fault f_1 appears, thus the set M_1 is an FMSO set. M_2 contains the constraint c_2 and can therefore be used to detect fault f_2 . M_3 contains c_2 and c_4 , thus $\mathcal{F}_{M_3} = \{f_1, f_2\}$.

The MSO set M_4 contains neither c_2 nor c_4 , which leads to $\mathcal{F}_{M_4} = \emptyset$ thus M_4 is not an FMSO set. M_4 is called a Clear Minimal Structurally Overdetermined (CMSO) set [Pér17]. But these are not of importance for the structural analysis made in this work.

With this, the basic definitions for structural analysis are summarized and the actual algorithms for the structural analysis are presented.

4.4 FMSO algorithm

The first step of finding an FMSO is to determine the over-constrained subgraph of the system or rather its set of vertices \mathcal{V}_X^+ and \mathcal{V}_C^+ [Pér17]. In MATLAB® this can be done with the command “`dmperm`” which is cited in the MATLAB® documentation from [PF90].

4.4.1 An algorithm for the Dulmage-Mendelsohn decomposition

The first step of the algorithm is to find a maximal matching for the given graph, but before explaining the algorithm the following term needs to be introduced.

An **augmenting path** is an alternating path that starts and ends with unmatched vertices. If the first unmatched vertex is from the set \mathcal{V}_X , the augmenting path needs to terminate with a vertex of \mathcal{V}_C and vice versa. [PF90]

The algorithm first looks for a maximum matching in the biadjacency matrix of the graph. In the following \mathcal{V}_C is the set of row vertices and \mathcal{V}_X the set of column vertices and a vertex is matched, if it is an element of an edge in \mathcal{M} .

The biadjacency matrix of the bipartite graph \mathcal{G} is an $m \times n$ matrix, where $m = |\mathcal{V}_C|$ are the number of rows and $n = |\mathcal{V}_X|$ the number of columns. In [PF90] it is proven that for $m \geq n$ the presented algorithm finds a maximum matching for the graph \mathcal{G} . If $m < n$, the transposed biadjacency matrix can be considered.

The algorithm 1 will be explained with the following example which is the same as in (4.15) but with reordered rows and columns so that in the end the results can be compared.

	x_8	x_1	x_3	x_5	x_4	x_6	x_2	x_7	x_9
c_9	X								X
c_5		X	X		X				
c_3		X							
c_7				X			X		
c_1		X					X		
c_6				X	X	X			
c_2							X		
c_4			X		X				
c_8	X			X		X		X	

Algorithm 1: Maximum Matching

```

% Step 1 (Cheap matching)
Initialize matching  $\mathcal{M}$  and the set of unmatched column vertices  $\mathcal{U}$  as an empty set;
foreach  $x \in \mathcal{V}_X$  do
    Match  $x$  to the first unmatched  $c \in \mathcal{V}_C$ , if possible;
    if  $x$  cannot be matched then Put  $x$  into  $\mathcal{U}$ ;
% Step 2 (Augment matching)
Initialize  $\mathcal{U}_{new} = \emptyset$ ;
repeat
    foreach  $x \in \mathcal{U}$  do
        Start with  $x$  searching an augmenting path, visiting only row vertices of  $\mathcal{V}_C$ 
        that have not been visited previously in this pass;
        Mark all row vertices reached while searching as visited;
        if An augmenting path is found then Augment  $\mathcal{M}$ ;
        else Put  $x$  into  $\mathcal{U}_{new}$ ;
     $\mathcal{U} = \mathcal{U}_{new}$ ;
     $\mathcal{U}_{new} = \emptyset$ ;
until no augmenting path is found in a pass;

```

The first step is to determine a so called “cheap matching” which is going through all the column vertices of \mathcal{V}_X and matching them to the next free row vertex in \mathcal{V}_C . Going from x_8 to x_9 from the left to the right, the matching shown in equation (4.20) is obtained.

	x_8	x_1	x_3	x_5	x_4	x_6	x_2	x_7	x_9
c_9	<u>(X)</u>								X
c_5		<u>(X)</u>	X		X				
c_3		X							
c_7				<u>(X)</u>			X		
c_1		X					<u>(X)</u>		
c_6				X	<u>(X)</u>	X			
c_2							X		
c_4			<u>(X)</u>		X				
c_8	X			X		<u>(X)</u>		X	

The unmatched column vertices \mathcal{U} are now x_7 and x_9 which are used in the second part, the augment matching, which is shown in figure 4.6. The left graph shows the state as in equation (4.20), the right graph, the state after the matching has been updated. All matched vertices are underlined and the edges of the matching are drawn in bold.

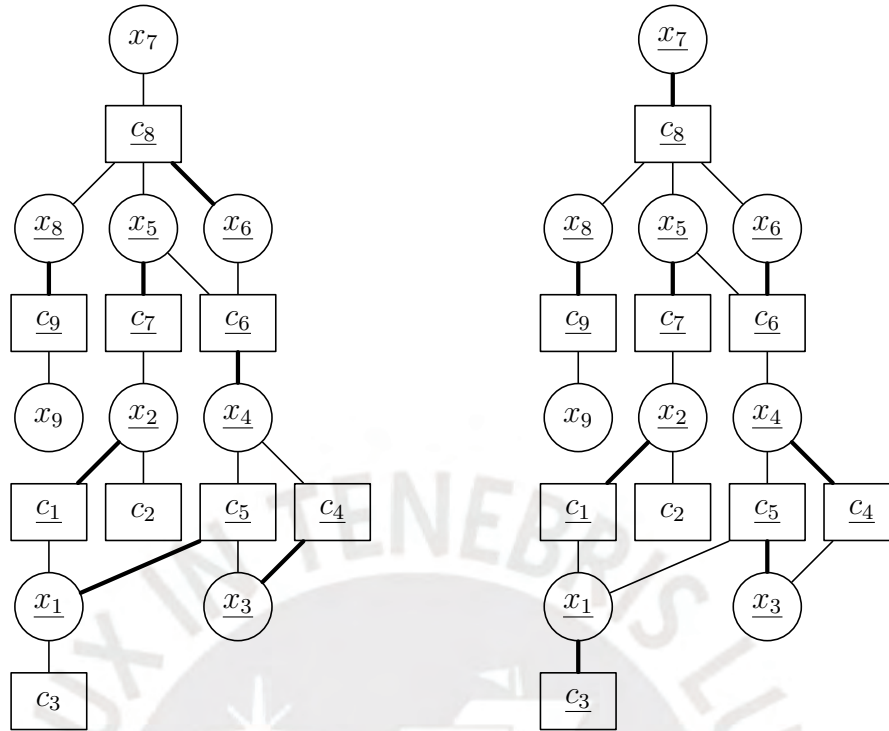


Fig. 4.6: Bipartite graph of equation (4.20) before and after augmenting the matching

The first step is to search for an augmenting path starting with x_7 . This can lead to $x_7, c_8, x_6, c_6, x_4, c_4, x_3, c_5, x_1, c_3$, and the path terminates with the unmatched row vertex c_3 so it is an augmenting path. Now the matching can be updated as follows.

The augmenting path starts with the unmatched edge $\{x_7, c_8\}$, therefore every second pair is a matched edge. Here $\{c_8, x_6\}, \{c_6, x_4\}, \{c_4, x_3\}$ and so on. To update the matching, the before unmatched edge $\{x_7, c_8\}$ is matched and the other matchings are corrected. Thus, now the matched edges are $\{x_7, c_8\}, \{x_6, c_6\}, \{x_4, c_4\}, \dots, \{x_1, c_3\}$, which can be seen on the right in figure 4.6.

Now there is one unmatched vertex in \mathcal{U} left, which is x_9 . Starting with x_9 , the path goes x_9, c_9, x_8 and terminates because when finding a path for x_7, c_8 had been visited already. As x_9, c_9, x_8 is not an augmenting path and there is no vertex in \mathcal{U} left and the inner loop of step 2 terminates. But an augmenting path was found so the `foreach` loop in step 2 is entered again and the set of visited vertices is cleared. Now the path is x_9, c_9, x_8, c_8, x_7 and terminates, because it is not possible to leave the vertex x_7 anymore as it is not connected to any other row vertex than c_8 . The inner loop is left and as no augmenting path was found, the algorithm terminates, thus a maximum matching with cardinality 8 was found. Considering that a matching does not need to be unique, the found maximum matching might not coincide with the maximum matching shown in (4.15), the original example. But it is a good sign, that the maximum matchings have the same cardinality.

Now the algorithm continues with decomposing the graph into the three subgraphs of the Dulmage-Mendelsohn decomposition or more exactly, determining the sets \mathcal{V}_X^+ , \mathcal{V}_X^0 , \mathcal{V}_X^- and \mathcal{V}_C^+ , \mathcal{V}_C^0 , \mathcal{V}_C^- , which can be done with the algorithm presented in 2. In [PF90] this is called the “coarse decomposition” because the graph can be decomposed further but this step is not shown in this work since it is not necessary for finding FMSOs.

Algorithm 2: Coarse Decomposition

```

% Step 0 (Initialize)
Initialize  $\mathcal{V}_X^0 = \mathcal{V}_X$  and  $\mathcal{V}_C^0 = \mathcal{V}_C$ ;
Initialize  $\mathcal{V}_X^+ = \mathcal{V}_X^- = \emptyset$ ;
Initialize  $\mathcal{V}_C^+ = \mathcal{V}_C^- = \emptyset$ ;
Set  $\mathcal{U}$  the set of unmatched column vertices left from algorithm 1;
Set  $\mathcal{C}$  the set of unmatched row vertices left from algorithm 1;
% Step 1 (Find  $\mathcal{V}_X^-$  and  $\mathcal{V}_C^-$ )
foreach  $x \in \mathcal{V}_X$  do
  Put  $x$  from  $\mathcal{V}_X^0$  in  $\mathcal{V}_X^-$ ;
  Put all row vertices reachable by an alternating path from  $x$ , from  $\mathcal{V}_C^0$  in  $\mathcal{V}_C^-$  and
  put all column vertices reachable by an alternating path from  $x$ , from  $\mathcal{V}_X^0$  in  $\mathcal{V}_X^-$ ;
% Step 2 (Find  $\mathcal{V}_X^+$  and  $\mathcal{V}_C^+$ )
foreach  $c \in \mathcal{C}$  do
  Put  $c$  from  $\mathcal{V}_C^0$  in  $\mathcal{V}_C^+$ ;
  Put all column vertices reachable by an alternating path from  $c$ , from  $\mathcal{V}_X^0$  in  $\mathcal{C}^+$ 
  and put all row vertices reachable by an alternating path from  $c$ , from  $\mathcal{V}_C^0$  in  $\mathcal{V}_C^+$ ;

```

Starting with the unmatched column from algorithm 1, an alternating path is x_9, c_9, x_8, c_8, x_7 .

Thus $\mathcal{V}_X^- = \{x_7, x_8, x_9\}$ and $\mathcal{V}_C^- = \{c_8, c_9\}$.

The only unmatched row vertex is c_2 . An alternating path from c_2 is c_2, x_2, c_1, x_1, c_3 , thus $\mathcal{V}_X^+ = \{x_1, x_2\}$ and $\mathcal{V}_C^+ = \{c_1, c_2, c_3\}$.

The vertices not in a set yet yield to $\mathcal{V}_X^0 = \{x_3, x_4, x_5, x_6\}$ and $\mathcal{V}_C^0 = \{c_4, c_5, c_6, c_7\}$.

The obtained result is the same as in the example in equation (4.15), thus the next step is to find the MSOs in \mathcal{G}^+ .

4.4.2 An algorithm for finding FMSOs

Finding FMSOs means to first find the MSOs of \mathcal{G}^+ . The algorithm presented here was taken from [KÅN05] where two algorithms are presented. Here the simple one is shown to get the main idea and afterwards the idea of the improved one.

The algorithm expects as an input the biadjacency matrix of \mathcal{G}^+ obtained by the DM decomposition explained in the previous section 4.4.1. The main idea is to simply take

away one constraint of the system, then recalculate all PSOs and check them for structural redundancy of one. If a PSO set has a structural redundancy of one, it is an MSO and therefore stored.

The algorithm 3 is a recursive function which returns the set of MSOs found in \mathcal{G}^+ . The

Algorithm 3: $\mathcal{M}_{MSO} = \text{FindMSO}(M)$ (simple)

```

 $\rho = |\mathcal{V}_C(M)| - |\mathcal{V}_X(M)|;$ 
if  $\rho = 1$  then
   $\mathcal{M}_{MSO} = \{M\};$ 
else
   $\mathcal{M}_{MSO} = \emptyset;$ 
  foreach  $c \in M$  do
     $M' = (M \setminus \{c\})^+;$ 
     $\mathcal{M}_{MSO} = \mathcal{M}_{MSO} \cup \text{FindMSO}(M');$ 
return  $\mathcal{M}_{MSO}$ 

```

variable ρ is the structural redundancy of the input matrix M . $|\mathcal{V}_C(M)|$ are then the number of rows of M and $|\mathcal{V}_X(M)|$ the number of columns. The expression $M' = (M \setminus \{c\})^+$ means, store the PSO part of the biadjacency matrix $(M \setminus \{c\})$ in M' , which can be accomplished by applying the DM decomposition explained in section 4.4.1 to the matrix $(M \setminus \{c\})$. M' is the biadjacency matrix of the graph $\mathcal{G}(\mathcal{V}_C^+(M \setminus \{c\}) \cup \mathcal{V}_X^+(M \setminus \{c\}), \mathcal{E})$ and eliminates the case, that a graph can be structurally overdetermined but not every vertex can be reached by an alternating path from a free vertex in \mathcal{V}_C^+ as explained with the graph \mathcal{G}^2 in figure 4.5. [KÅN05]

The algorithm 3 finds all MSOs but has the problem that some MSOs are found multiple times.

Therefore, so called **equivalence classes** are introduced. In the following the notion, where $M' = (M \setminus \{c\})^+$ is the graph of a PSO set, is used to explain what is an equivalence class. In figure 4.7 the steps for the simple algorithm 3 are shown. The algorithm is applied to the graph of equation (4.18) which can be seen as 0) in the picture. The found PSO sets, which are also MSO sets, are marked gray.

It can be seen that for removing the vertices c_1 or c_2 , the same PSO set is found, namely $(M \setminus \{c_1\})^+ = (M \setminus \{c_2\})^+ = \{c_3, c_4\}$.

The idea is to merge c_1 and c_2 into a so called equivalence class E_i [KÅN05]. This can be imagined as merging the subgraph with the vertices c_1 and c_2 into a single vertex E_i . The equivalence classes of the example would be $E_1 = \{c_1, c_2\}$, $E_2 = \{c_3\}$ and $E_3 = \{c_4\}$, as shown in figure 4.8. It can then be searched for MSOs in the new bipartite graph with the

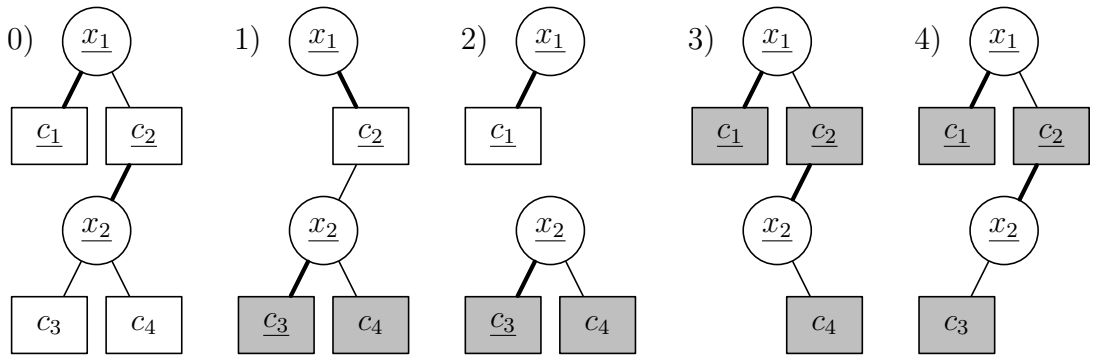


Fig. 4.7: Steps of the simple algorithm for finding MSOs

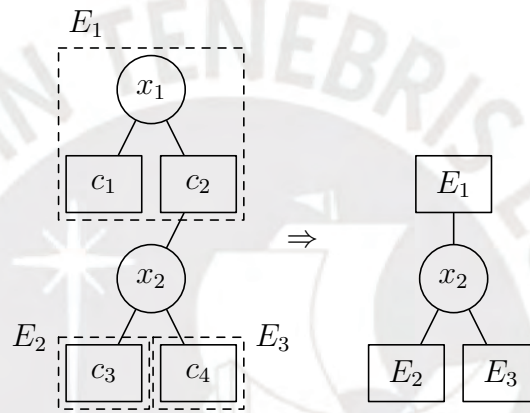


Fig. 4.8: Example of building equivalence classes in a bipartite graph

equivalence classes instead of the whole graph.

Note that an equivalence does not necessarily contain more than one vertex, as seen in E_2 . These equivalence classes can be found by starting the algorithm with every single vertex of \mathcal{V}_C as an equivalence class. The set of all equivalence classes is E . Then equivalence classes can be found by calculating $(E \setminus \{E_i\})^+$ and $(E \setminus \{E_i\})^0$. Two equivalence classes can be merged into one, if $(E \setminus \{E_i\})^0 \neq \emptyset$ where $(E \setminus \{E_i\})^0$ is the just constrained part of the graph with the equivalence classes. Consequently, the new equivalence class is $E_{new} = E_i \cup (E \setminus \{E_i\})^0$. [KÅN05]

The complete theory behind this and the improved algorithm is also presented in [Kry06] and an implementation for MATLAB® is provided via the free toolbox presented in [FKJ17].

Once all MSOs are found, every MSO needs to be evaluated if it is an FMSO by checking if the MSO contains constraints with faults and if yes which faults are included in the FMSO. [Pér17]

Note that there exist a lot more algorithms than the ones presented here, used for structural analysis, a comparison of some of the algorithms and the results can be found in [ABE⁺09] or [Pér17]. Whereas the latter one focuses more on the fault diagnosis of distributed systems and presents ideas about the FMSO selection when the information exchanged by the distributed systems has certain restrictions.

With that the algorithm used for the structural analysis in this work is presented and the structural analysis of the RO system can be done.

4.5 Structural analysis of the RO system

Before the structural model of the RO plant is given, the number of constraints is reduced by eliminating variables which can be substituted easily. This is the motor torque τ_p which is not measured anyway and just appears in the differential equation of the rotational speed of the pump. Furthermore, the variable Q_m is eliminated, as it is not affected by a fault and therefore always holds $Q_m = Q_p$. This reduces the number of variables and constraints by two and makes the handling of the equation system easier. Moreover, the constant parameters of the model are not taken into account as variables since they are considered as known and static. If a parameter is unknown, it could be considered as an unknown variable for the structural analysis.

Of course even more variables could be eliminated but it might not be necessary and difficult. Beside that, it is required that a single fault just violates one equation. If it does not, a new variable, for example, $x_f = f$ needs to be introduced which leaves the fault in a single equation. This needs to be done because otherwise it can occur that a fault needs to be eliminated like an unknown variable and therefore does not appear in the ARR, even though it appears in the constraints of the FMSO [EBP⁺19].

Furthermore, one could easily substitute the measured variables without the faults in c_{13}, c_{15}, \dots . But substituting the unknown variable within all the constraints through its directly measured value takes the possibility that an unknown variable can be also determined by a combination of other measured variables as seen later when calculating the ARRs for the RO system.

The resulting equation system can be seen in (4.20).

$$\begin{array}{cccccccccccccccc}
x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} \\
Q_f & Q_p & Q_b & C_f & C_m & C_b & C_p & C_{ms} & \Delta P & \omega_p & u_p & u_v & A_v & \gamma_p & \gamma_f & pH
\end{array}$$

$$\begin{aligned}
c_1 : x_1 &= x_2 + x_3 \\
c_2 : \dot{x}_6 &= \frac{1}{V_b}(x_1x_4 - x_6x_3 - x_2x_5) \\
c_3 : \dot{x}_7 &= \frac{1}{V_p}x_2(x_5 - x_7) \\
c_4 : x_2 &= k_m(A_m - f_2)(x_9 - \frac{1}{\beta}(x_8 - x_5)) \\
c_5 : x_5 &= \frac{1}{2}\left(x_8 - \beta x_9 - \alpha + \sqrt{\beta^2 x_9^2 - 2\beta x_9(x_8 - \alpha) + (x_8 + \alpha)^2}\right) \\
c_6 : x_8 &= \frac{1}{2}(x_4 + x_6) \\
c_7 : x_1 &= \frac{30V_d}{\pi}x_{10} - f_1 \\
c_8 : \dot{x}_{10} &= \frac{1}{J_p}\left(-x_9 \frac{5 \cdot 10^4 V_d}{\pi} - dx_{10} + c(-n_p x_{10} + \omega_{el}^* x_{11}) - f_3\right) \\
c_9 : x_3 &= \nu x_{13} \sqrt{x_9} \\
c_{10} : \dot{x}_{13} &= \frac{1}{\tau_v}(-x_{13} + A_v^* x_{12} + f_4) \\
c_{11} : x_{14} &= b_{p0}(x_7 - C_p^*) - b_{p1}(x_{16} - pH^*) + \gamma_p^* \\
c_{12} : x_{15} &= b_{f0}(x_4 - C_f^*) - b_{f1}(x_{16} - pH^*) + \gamma_f^*
\end{aligned} \tag{4.20}$$

Measurements:

$$\begin{aligned}
c_{13} : y_1 &= x_{16} \\
c_{14} : y_2 &= x_2 + f_5 \\
c_{15} : y_3 &= x_3 \\
c_{16} : y_4 &= x_9 \\
c_{17} : y_5 &= x_{14} + f_6 \\
c_{18} : y_6 &= x_{15}
\end{aligned}$$

Inputs :

$$\begin{aligned}
c_{19} : y_7 &= x_{11} \\
c_{20} : y_8 &= x_{12}
\end{aligned}$$

Based on this model the bi-adjacency matrix is shown in table 4.1

Table 4.1: Biadjacency matrix of the reduced RO plant

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	f_1	f_2	f_3	f_4	f_5	f_6	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8		
c_1	X	X	X																													
c_2	X	X	X	X	X	X																										
c_3		X			X		X																									
c_4		X			X			X	X									X														
c_5					X			X	X																							
c_6				X		X		X																								
c_7	X									X							X															
c_8									X	X	X								X													
c_9			X						X				X																			
c_{10}												X	X							X												
c_{11}							X							X		X																
c_{12}				X											X	X																
c_{13}																X							X									
c_{14}		X																			X			X								
c_{15}			X																							X						
c_{16}									X																		X					
c_{17}													X									X						X				
c_{18}															X														X			
c_{19}											X																			X		
c_{20}												X																			X	

The DM decomposition of the system of the reduced matrix (just the columns with unknowns) of 4.1 reveals that $\mathcal{V}_X = \mathcal{V}_X^+$. Thus, the whole set of constraints is a PSO set and the structural redundancy is 4. The next algorithm finds then 73 MSOs and as every MSO contains at least one constraint with a fault, there are 73 FMSOs which can be all reviewed in the appendix A. The sets are given in the form $\varphi \subseteq \mathcal{V}_C$ and $\mathcal{F}_\varphi \subseteq \mathcal{F}$, e.g. $\varphi_1 = \{c_9, c_{10}, c_{15}, c_{16}, c_{20}\}$ and $\mathcal{F}_{\varphi_1} = \{f_4\}$ or $\varphi_2 = \{c_3, c_4, c_5, c_{11}, c_{13}, c_{14}, c_{16}, c_{17}\}$ and $\mathcal{F}_{\varphi_2} = \{f_2, f_5, f_6\}$.

4.5.1 FMSO selection

As there are many FMSOs with the same set of faults, for example, there are four FMSOs with $\mathcal{F}_\varphi = \{f_2, f_5, f_6\}$. The first step is to find all different sets of \mathcal{F}_φ which can be made through the fault signature matrix. The fault signature matrix, from now on F_{SM} , is a biadjacency matrix with one set of vertices, the FMSOs, and the other set with the faults \mathcal{F} . If the FMSOs are the rows, one row can be seen as the fault signature of an FMSO. Then for the RO system, the fault signature matrix looks as in equation (4.21).

	f_1	f_2	f_3	f_4	f_5	f_6
φ_1				X		
φ_2		X			X	X
φ_3		X		X	X	X
φ_4	X	X	X		X	
\vdots				\vdots		
φ_{73}	X	X	X	X		X

(4.21)

Replacing the fields with an “X” through 1 and the empty fields with 0, leads to a binary matrix where a row can be seen as a binary number with 6 bits. Thus, the first row would be the binary number 000100b. Seeing the binary number as a row vector and every digit as the column of a vector, the decimal value of this number can be calculated by multiplying the vector with $v_b = [2^0, 2^1, 2^2, 2^3, 2^4, 2^5]^T$.

Multiplying the complete fault signature matrix in binary form with this vector leads to $v_{\varphi S} = F_{SM} \cdot v_b$ as shown in equation (4.22).

$$\begin{bmatrix} 8 \\ 50 \\ 58 \\ 23 \\ \vdots \\ 47 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ & & & \vdots & & \\ 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2^0 \\ 2^1 \\ 2^2 \\ 2^3 \\ 2^4 \\ 2^5 \end{bmatrix} \quad (4.22)$$

$v_{\varphi S}$ can then be seen as the fault signature of an FMSO and the number of different elements in the vector $v_{\varphi S}$ is the number of different fault signatures.

The vector $v_{\varphi S}$ for the found FMSOs has 23 different elements, thus there are 23 different fault signatures which are shown in equation 4.2. n_{FMSOs} is the number of FMSOs with the according fault signature. With the fault signature matrix it can be evaluated if a fault is

Table 4.2: All different fault signatures of the found FMSOs

	f_1	f_2	f_3	f_4	f_5	f_6	n_{FMSOs}		f_1	f_2	f_3	f_4	f_5	f_6	n_{FMSOs}
1				X			(1)	13	X		X	X	X		(2)
2		X			X	X	(4)	14		X			X		(1)
3		X		X	X	X	(6)	15		X		X	X		(2)
4	X	X	X		X		(3)	16	X	X	X				(1)
5	X	X	X	X	X		(3)	17	X	X	X	X			(2)
6	X		X		X	X	(3)	18					X	X	(1)
7	X		X	X	X	X	(3)	19				X	X	X	(2)
8	X	X	X		X	X	(9)	20	X		X			X	(1)
9	X	X	X	X	X	X	(8)	21	X		X	X		X	(2)
10	X	X	X			X	(6)	22		X				X	(1)
11	X	X	X	X		X	(9)	23		X		X		X	(2)
12	X		X		X		(1)								

detectable and isolable. It is not necessary to reduce it before, but it is easier to handle 23 than 73 signatures. The following definitions taken from [PCT⁺18] state when a fault is detectable and when a fault is isolable.

Definition 4.5.1. (Detectable fault) [PCT⁺18]

A fault $f \in \mathcal{F}$ is detectable in the system $\Sigma(y, x, f)$ if there is an FMSO set φ such that $f \in \mathcal{F}_\varphi$, where the set of faults of the system is denoted by \mathcal{F} and \mathcal{F}_φ is the set of faults appearing in the FMSO φ .

Definition 4.5.2. (Isolable fault) [PCT⁺18]

Given two detectable faults f_j and f_k of \mathcal{F} , $j \neq k$, f_j is isolable from f_k if there exists an FMSO set φ such that $f_j \in F_\varphi$ and $f_k \notin F_\varphi$

In terms of the fault signature matrix, definition 4.5.1 means that if there is a column v_i of F_{SM} where $\|v_i\| = 0$, then the fault f_i is not detectable because the fault does not appear in any of the FMSOs. With $\|v_i\|$ any norm of the column vector v_i . In table 4.2 there is no empty column, thus every fault is detectable.

If a fault is not detectable, it is not isolable from another fault either. But if two faults are detectable then it can be checked with $\|v_i - v_j\|$ if they are isolable from each other. With v_x the column of the faultmatrix for fault f_x , $\|v_i - v_j\|$ determines if two columns are equal. They are equal if $\|v_i - v_j\| = 0$ and not equal otherwise. If they are equal, it means that every time both, fault f_i and also fault f_j appear in the FMSO, which means that they are not isolable from each other. This case indeed appears in table 4.2, namely for fault f_1 and f_3 .

In order to find a minimal set of FMSOs, the matrix in 4.2 can be further reduced by the column 1 or 3. Finding a minimal set of FMSOs for maximal fault detection and isolation depends on the assumptions made and the requirements of the fault diagnosis system [Pér17]. Note that even though the fault matrix was reduced, the set of FMSOs is still the same because when a certain fault signature is selected, the FMSOs which have this fault signature can still be more than one. Calculating the residuals for all 73 FMSOs is not an option, but the selection can be reduced by making some considerations. For selecting a minimal number of FMSOs, first the notion of analytical redundancy relations needs to be introduced to understand why to choose a certain FMSO.

Definition 4.5.3. (Analytical redundancy relation) [PCT⁺17]

For a system $\Sigma(x, y, f)$, with x the unknown variables of the system, y the known variables and f its faults to be evaluated. The relation $arr(y, \dot{y}, \ddot{y}, \dots) = 0$ is an **Analytical Redundancy Relation** (ARR) for $\Sigma(x, y, f)$ if for each y consistent with $\Sigma(x, y, f)$ the relation is fulfilled. The scalar function $arr(y, \dot{y}, \ddot{y}, \dots)$ is also called a residual generator.

It is assumed, that an FMSO set can be used to deduce such an ARR. An ARR is seen as the causal interpretation of a (F)MSO set. Problems when deducing an ARR from an FMSO can occur with so called differential loops which shown with the following problem.

$$\begin{aligned}
c_1 : 0 &= -\dot{x}_1(t) - x_1(t) + x_2(t) + f_1(t) \\
c_2 : 0 &= -\dot{x}_1(t) + y_1(t) \\
c_3 : 0 &= -x_2(t) + y_2(t)
\end{aligned}$$

	x_1	x_2	f_1
c_1	X	X	X
c_2	X		
c_3		X	

(4.23)

An FMSO set is $\{c_1, c_2, c_3\}$ and deducing the corresponding ARR would be as follows.

Replacing $\dot{x}_1(t)$ from c_2 and $x_2(t)$ from c_3 in c_1 leads to $0 = -y_1(t) - x_1(t) + y_2(t) + f_1(t)$. $x_1(t)$ can be obtained by integrating $y_1(t)$, it holds $x_1(t) - x_1(t_0) = \int_{t_0}^t y_1(\tau) d\tau$ thus $x_1(t) = \int_{t_0}^t y_1(\tau) d\tau + x_1(t_0)$. The problem is now that the constant $x_1(t_0)$ might be unknown. If the system is fault free, then $f_1(t) = 0$, thus $arr(y(t)) = -y_1(t) - \int_{t_0}^t y_1(\tau) d\tau - x_1(t_0) + y_2(t)$ should be zero. y_1 and y_2 can be consistent with the system but $arr(y(t)) \neq 0$ because of a wrong value for $x_1(t_0)$. If the analytical redundancy relation $arr(y(t)) = 0$ is not fulfilled, it will be wrongly assumed, that the system is faulty. [BKL⁺16]

Another problem might occur with non-linear constraints like in following the system in (4.24).

$$\begin{aligned}
c_1 : 0 &= -x_1 + x_2 + f_1 \\
c_2 : 0 &= -|x_1| + y_1 \\
c_3 : 0 &= -x_2 + y_2
\end{aligned}$$

	x_1	x_2	f_1
c_1	X	X	X
c_2	X		
c_3		X	

(4.24)

The residual $-y_1 + y_2$ then just holds if $x_1 \geq 0$.

Selecting FMSOs for systems described by non-linear and/or differential constraints assumes therefore always a best case scenario [KÅF10].

Assuming that if the system is faulty, this can be traced back to one occurred fault, the minimal number n_{minF} of FMSOs would be [Pér17]:

$$n_{minF} = \lceil \log_2(|\mathcal{F}_{isol}| + 1) \rceil \quad (4.25)$$

Where $\lceil \cdot \rceil$ denotes the ceiling function and \mathcal{F}_{isol} the set of isolable faults. Note that a binary number with n digits has 2^n different bit pattern, the zero included. However, equation

(4.25) excludes zero as a pattern.

Taking a system with 3 faults, a possible fault signature matrix would be:

	f_1	f_2	f_3
1	X	X	
2		X	X
3	X		X

(4.26)

With equation (4.25) the minimum number of FMSOs would be 2, which can be accomplished with these fault signatures. Considering that an ARR is just violated if a fault is active and only one fault at a time occurs, it is sufficient to evaluate the violated ARRs to determine which fault is active. The fault signature matrix therefore indicates which of the ARRs are violated if the fault f_i is active [Pér17]. In the case of example (4.26) every pair of FMSOs can be taken because it can always be distinguished which fault is active. Taking the FMSOs with the fault signature of row 1 and row 2, the fault f_1 is active if the residual generator of the according FMSO is unequal to zero. For f_2 both residual generators would be unequal to zero and for f_3 the other one of the two.

For the RO system the number of isolable faults would be five as fault f_1 and f_3 are not isolable from each other. Therefore the faults are considered as one isolable fault as they are still isolable from the other ones. Thus, the minimum number of FMSOs would be three. FMSO selection can be made via existing algorithms. For example, the one presented in [PCT⁺18] selects the FMSOs for the distributed case. For distributed systems, the requirements according to information exchange need to be considered. Whereas in a global approach such limits do not occur. Nevertheless, the idea presented in [PCT⁺18] can also be applied to the global approach.

However, the FMSO selection in this work was made manually by first choosing the three FMSOs with the minimum number of constraints which are φ_1 , φ_2 and φ_{21} . This leads to the fault signatures in equation (4.27) where the row numbers are the row numbers of the matrix in 4.2.

	f_1	f_2	f_3	f_4	f_5	f_6
1				X		
2		X			X	X
12	X		X		X	

(4.27)

Just the fault f_4 appears isolated from all the other faults in a single FMSO. Thus, it is first not further considered for the FMSO selection. With rows two and three the other faults

can be detected but not all can be isolated from each other. Fault f_2 and f_6 just appear in row two, therefore they cannot be isolated from each other. The faults f_1 and f_3 anyway cannot be isolated from each other. Nevertheless, fault f_5 appears in row two and three, it can be isolated from f_1 and f_3 and also from f_2 and f_6 . Therefore, one additional FMSO is needed which includes either f_2 or f_6 but not both. Looking in the fault signature matrix of table 4.2, these are the rows 4 - 7 and 14 - 21. Choosing row 14 where the faults f_2 , f_4 and f_5 appear makes it even possible to eliminate the first row and three signatures for a minimum set of FMSOs are determined as seen in the matrix in (4.28).

	f_1	f_2	f_3	f_4	f_5	f_6	n_{FMSOs}
2		X			X	X	(4)
3	X		X		X		(1)
15		X		X	X		(2)

Note that for maximal isolability every column needs to have a different pattern. If these pattern are again seen as binary numbers then for 3 FMSOs there are 7 different binary numbers excluding the pattern 0b000. The fault f_1 has the number 0b010, f_2 0b101 and so on. With this, a selection of FMSOs has been found, which are characterized by its fault signature. There are still 7 FMSOs which lead to the desired fault signatures. Choosing again the ones with the minimal number of constraints, leads to the FMSOs shown in (4.29).

φ	\mathcal{F}_φ
2. $\{c_3, c_4, c_5, c_{11}, c_{13}, c_{14}, c_{16}, c_{17}\}$	$\{f_2, f_5, f_6\}$
21. $\{c_1, c_7, c_8, c_{14}, c_{15}, c_{16}, c_{19}\}$	$\{f_1, f_3, f_5\}$
30. $\{c_1, c_2, c_4, c_5, c_6, c_9, c_{10}, c_{12}, c_{13}, c_{14}, c_{16}, c_{18}, c_{20}\}$	$\{f_2, f_4, f_5\}$

In the sense of structural analysis for the RO system analyzed in this work, it can be said that all the faults that are considered can be detected. However, the faults f_1 which is the leakage of the feed stream and f_3 the motor torque fault cannot be isolated from each other. If it is necessary that these two faults are isolable from each other, it can be evaluated if measuring another variable or searching for new constraints which give more information about the coupling of these two faults would lead to other FMSOs.

Note that until now just a minimal selection of FMSOs is made, which makes it possible, in the sense of structural analysis, to detect and isolate the maximum number of faults. For an implementation of the fault diagnoser it is still necessary to deduce the ARRs from the chosen FMSOs, which will be done in the next chapter.

Implementation proposal

In this chapter the ARR_s are deduced from the selected FMSOs determined by the structural analysis of the RO system. After the ARR_s are calculated, the residuals are simulated with the model proposed in chapter 3. With that finishes the offline design of the fault diagnosis system. The proposed residuals can be used for online fault detection and isolation in a fault diagnosis system of the RO plant. The structure of the implementation can be seen in figure 5.1. The measured variables are first pre-processed. The pre-processing depends mainly on the measurement noise of the measured variables. Here filtering the outputs might be required. Furthermore the necessary derivatives of the measured variables need to be calculated. An approach for that will be shown in section 5.2. Based on the outputs of the residuals online fault detection and isolation can be made.

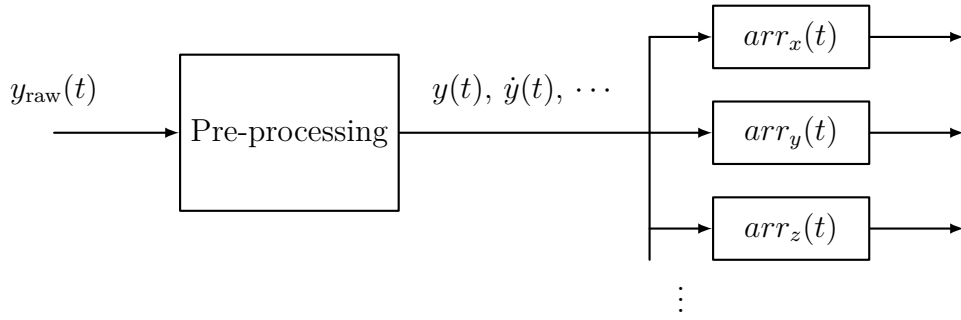


Fig. 5.1: Structure of the online fault diagnosis system

5.1 Deducing the ARRrS

5.1.1 ARR of FMSO 2

The first step is to reduce the model taken for structural analysis by the equations which contain the measured variables y_i . Hence, starting with $\varphi_2 = \{c_3, c_4, c_5, c_{11}, c_{13}, c_{14}, c_{16}, c_{17}\}$ and substituting the known variables from $c_{13}, c_{14}, c_{16}, c_{17}$, the constraints left are:

$$\begin{aligned}
 c_3 : \quad \dot{x}_7 &= \frac{1}{V_p}(y_2 - f_5)(x_5 - x_7) \\
 c_4 : \quad (y_2 - f_5) &= k_m(A_m - f_2)(y_4 - \frac{1}{\beta}(x_8 - x_5)) \\
 c_5 : \quad x_5 &= \frac{1}{2}(x_8 - \beta y_4 - \alpha + \sqrt{\beta^2 y_4^2 - 2\beta y_4(x_8 - \alpha) + (x_8 + \alpha)^2}) \\
 c_{11} : \quad (y_5 - f_6) &= b_{p0}(x_7 - C_p^*) - b_{p1}(y_1 - pH^*) + \gamma_p^*
 \end{aligned}$$

At first, the constraints with only one remaining unknown variable are taken, which is just c_{11} . From c_{11} the following relation can be deduced:

$$x_7 = \frac{1}{b_{p0}} \underbrace{[y_5 - f_6 - \gamma_p^* + b_{p1}(y_1 - pH^*)]}_{\alpha_1} + C_p^* \quad (5.1)$$

Now x_7 can be replaced in every constraint where it appears. This is just c_3 , but c_3 also includes the time derivative \dot{x}_7 . Assuming that the expression given for x_7 is differentiable with respect to time, \dot{x}_7 can be expressed as:

$$\dot{x}_7 = \frac{1}{b_{p0}} \underbrace{[\dot{y}_5 - \dot{f}_6 + b_{p1}\dot{y}_1]}_{\dot{\alpha}_1} \quad (5.2)$$

Now x_7 and \dot{x}_7 can be replaced in c_3 leading to:

$$x_5 = \frac{V_p}{y_2 - f_5} \dot{a}_1 + a_1 \quad (5.3)$$

With a_1 and \dot{a}_1 from the equations (5.1) and (5.2).

The remaining constraints which have not been used yet are c_4 and c_5 . Substituting x_5 in c_4 gives:

$$x_8 = \beta \left[\frac{f_5 - y_2}{k_m(A_m - f_2)} + y_4 \right] + \frac{V_p}{y_2 - f_5} \dot{a}_1 + a_1 \quad (5.4)$$

Now x_5 and x_8 can be replaced in c_5 , which would lead to an expression with a root. Considering the physical background, namely calculating the concentration $C_m = x_5$ for a given concentration of the membrane surface $C_{ms} = x_8$ and a given pressure $\Delta P = y_4$, another form of constraint c_5 can be taken. For calculating the ARR it is easier to take the expression shown in equation (3.10) which is in terms of x and y :

$$x_5^2 + x_5(\alpha + \beta y_4 - x_8) - \alpha x_8 = 0 \quad (5.5)$$

Substituting x_5 and x_8 in (5.5) leads to the first ARR

$$\begin{aligned} & \left(\frac{V_p}{y_2 - f_5} \dot{a}_1 + a_1 \right)^2 + \\ & \left(\frac{V_p}{y_2 - f_5} \dot{a}_1 + a_1 \right) \cdot \left(\alpha + \beta y_4 - \beta \left[\frac{f_5 - y_2}{k_m(A_m - f_2)} + y_4 \right] - \frac{V_p}{y_2 - f_5} \dot{a}_1 - a_1 \right) - \\ & \alpha \left(\beta \left[\frac{f_5 - y_2}{k_m(A_m - f_2)} + y_4 \right] + \frac{V_p}{y_2 - f_5} \dot{a}_1 + a_1 \right) \\ & = 0 \end{aligned} \quad (5.6)$$

Which can be further simplified to:

$$\begin{aligned} & \frac{1}{b_{p0}k_m(A_m - f_2)} \left[V_p (\dot{y}_5 - \dot{f}_6 + b_{p1}\dot{y}_1) + (y_2 - f_5) (y_5 - f_6 + b_{p1}y_1 + a_2) \right] \\ & - \alpha y_4 = 0 \end{aligned} \quad (5.7)$$

with:

$$a_2 = b_{p0} (C_p^* + \alpha) - \gamma_p^* - b_{p1}pH^*$$

It can be seen that if $A_m = f_2$ which is the case when the membrane is 100% clogged the ARR does not have a solution. Furthermore, if $f_5 = y_2$ the absolute value of f_6 has no influence anymore, which was excluded by saying that just one fault appears at a time. The

residual generator in equation (5.7) is from now on referred to as arr_{256} .

5.1.2 ARR of FMSO 21

For FMSO φ_{21} , constraints $\{c_1, c_7, c_8, c_{14}, c_{15}, c_{16}, c_{19}\}$ are used to deduce the second ARR leading to the already reduced constraints.

$$\begin{aligned} c_1 : x_1 &= (y_2 - f_5) + y_3 \\ c_7 : x_1 &= \frac{30V_d}{\pi} x_{10} - f_1 \\ c_8 : \dot{x}_{10} &= \frac{1}{J_p} (-y_4 \frac{5 \cdot 10^4 V_d}{\pi} - dx_{10} + c(-n_p x_{10} + \omega_{el}^* y_7) - f_3) \end{aligned}$$

Substituting x_1 from c_1 in c_7 gives:

$$x_{10} = \frac{\pi}{30V_d} (y_2 + y_3 + f_1 - f_5) \quad (5.8)$$

The time derivative of x_{10} is then:

$$\dot{x}_{10} = \frac{\pi}{30V_d} (\dot{y}_2 + \dot{y}_3 + \dot{f}_1 - \dot{f}_5) \quad (5.9)$$

x_{10} and \dot{x}_{10} substituted in c_8 leads to the second ARR shown in equation (5.10).

$$\begin{aligned} &\frac{\pi J_p}{30V_d} (\dot{y}_2 + \dot{y}_3 + \dot{f}_1 - \dot{f}_5) + \frac{\pi(d + cn_p)}{30V_d} (y_2 + y_3 + f_1 - f_5) + \\ &\frac{5 \cdot 10^4 V_d}{\pi} y_4 - cw_{el}^* y_7 + f_3 = 0 \end{aligned} \quad (5.10)$$

Trying to get all the terms which contain faults on one side, the form shown in equation (5.11) can be obtained, which gives the opportunity to receive even more information from the output of the residual. Assume that it is known that just fault f_3 is active and there are no other faults in the real system than the ones considered. Then the residual arr_{135} (the right side of equation (5.11)) outputs directly the value of f_3 , which is not just fault isolation but also determining the magnitude of the fault.

$$\begin{aligned} &f_3 + \frac{\pi}{30V_d} (J_p(\dot{f}_1 - \dot{f}_5) + (d + cn_p)(f_1 - f_5)) \\ &= -\frac{\pi}{30V_d} (J_p(\dot{y}_2 + \dot{y}_3) + (d + cn_p)(y_2 + y_3)) - \frac{5 \cdot 10^4 V_d}{\pi} y_4 + cw_{el}^* y_7 \end{aligned} \quad (5.11)$$

The constraints c_8 and the one given in (5.8) show that if the rotational speed x_{10} of the high pressure pump were given through a measurement, it would be possible to isolate fault f_1 and f_3 because then 2 ARR's can be deduced from these two expressions.

5.1.3 ARR of FMSO 30

Now the last ARR is derived from the FMSO

$\varphi_{30} = \{c_1, c_2, c_4, c_5, c_6, c_9, c_{10}, c_{12}, c_{13}, c_{14}, c_{16}, c_{18}, c_{20}\}$. Substituting the measured variables from $c_{13}, c_{14}, c_{16}, c_{18}, c_{20}$ leads to:

$$\begin{aligned}
 c_1 : \quad x_1 &= (y_2 - f_5) + x_3 \\
 c_2 : \quad \dot{x}_6 &= \frac{1}{V_b}(x_1 x_4 - x_6 x_3 - (y_2 - f_5)x_5) \\
 c_4 : \quad (y_2 - f_5) &= k_m(A_m - f_2)(y_4 - \frac{1}{\beta}(x_8 - x_5)) \\
 c_5 : \quad x_5 &= \frac{1}{2}(x_8 - \beta y_4 - \alpha + \sqrt{\beta^2 y_4^2 - 2\beta y_4(x_8 - \alpha) + (x_8 + \alpha)^2}) \\
 c_6 : \quad x_8 &= \frac{1}{2}(x_4 + x_6) \\
 c_9 : \quad x_3 &= \nu x_{13} \sqrt{y_4} \\
 c_{10} : \quad \dot{x}_{13} &= \frac{1}{\tau_v}(-x_{13} + A_v^* y_8 + f_4) \\
 c_{12} : \quad y_6 &= b_{f0}(x_4 - C_f^*) - b_{f1}(y_1 - p H^*) + \gamma_f^*
 \end{aligned}$$

Remember that deducing an ARR from a non-linear differential algebraic system might not lead to a solution in the end. The idea includes finding a path for how to eliminate every unknown variable one by one and put it in the end into one equation to obtain the ARR. One can here start with selecting the last constraint where all the unknown variables are substituted. In this case constraint c_2 is chosen to be the last one as it seems the most "difficult". From there one can start a kind of path backwards how to eliminate the variables in the constraint. Starting with $x_1 \rightarrow c_1 \rightarrow x_3 \rightarrow c_9 \rightarrow x_{13} \rightarrow c_{10}$ where the path stops. If there are any doubts about how the equation can be solved for a certain variable, at first evaluate how to do it.

In this case it might be c_{10} where the differential equation needs to be solved to get x_{13} . The problem is that for solving the differential equation an initial value $x_{13}(t_0)$ is required. But in this case some assumptions can be made to make it still possible to calculate an ARR. Firstly, the differential equation in c_{10} is a ordinary linear and stable differential equation. Setting $\dot{x}_{13} = 0$ leads to $x_{13}(t) = A_v^* y_8(t)$ (the fault is here not considered), which is a stable equilibrium. For a constant $A_v^* y_8(t)$ the solution converges to this point asymptotically for any initial value of $x_{13}(t_0)$. In theory this takes an infinite time, but for practical purposes it might be sufficient if its close enough.

x_{13} is the cross sectional area of the brine valve and is physically limited to $0 \leq x_{13} \leq A_v^*$. Furthermore, the valve can be steered independently, which means it can be placed to a desired value before starting the actual reverse osmosis with the high pressure pump. Assuming that at least in the beginning it is fault free a good guess for an initial value can be made. Beside that, once the process is running and steady-state the value y_8 needs no significant changes anymore so that the solution has time to converge to the real value. It can be therefore tried with this approach to solve the differential equation online with the given input y_8 and a good guessed initial value for $0 \leq x_{13}(t_0) \leq A_v^*$. It only needs to be considered that the ARR might show a faulty behavior, although it is not faulty. However, the error vanishes if no fault is active.

With the calculated solution \hat{x}_{13} , x_1 and x_3 can be determined from c_1 and c_9 . The next unknown in c_2 is x_4 , which can be expressed with c_{12} as:

$$x_4 = \frac{1}{b_{f0}} (b_{f1} (y_1 - pH^*) - \gamma_f^*) + C_f^* \quad (5.12)$$

The next one is x_6 which can be calculated with c_6 but therefore x_8 is required.

With the constraints c_4 and c_5 , the following expressions for x_5 and x_8 can be obtained:

$$x_5 = \alpha \left(\frac{y_4 k_m (A_m - f_2)}{y_2 - f_5} - 1 \right) \quad (5.13)$$

$$x_8 = \frac{k_m (A_m - f_2)}{y_2 - f_5} - \frac{\beta (y_2 - f_5)}{k_m (A_m - f_2)} + \beta y_4 - \alpha \quad (5.14)$$

Now x_8 and x_4 can be substituted in c_6 to acquire x_6 :

$$x_6 = 2 \left(\frac{k_m (A_m - f_2)}{y_2 - f_5} - \frac{\beta (y_2 - f_5)}{k_m (A_m - f_2)} + \beta y_4 - \alpha \right) - \frac{1}{b_{f0}} (b_{f1} (y_1 - pH^*) - \gamma_f^*) - C_f^* \quad (5.15)$$

Calculating the time derivative of x_6 leads to:

$$\begin{aligned} \dot{x}_6 = 2 \left(\beta \dot{y}_4 - \frac{\beta (\dot{y}_2 - \dot{f}_5)}{k_m (A_m - f_2)} - \frac{k_m \dot{f}_2}{y_2 - f_5} - \frac{\beta (y_2 - f_5) \dot{f}_2}{k_m (A_m - f_2)^2} \right) - \\ \frac{2k_m (A_m - f_2) (\dot{y}_2 - \dot{f}_5)}{(y_2 - f_5)^2} - \frac{b_{f1}}{b_{f0}} \dot{y}_1 \end{aligned} \quad (5.16)$$

Consecutively, all the unknown variables of c_2 are expressed as terms of y_x . In this case it has not been merged into one equation but every value was calculated one by one and later substituted in:

$$0 = \frac{1}{V_b}(x_1x_4 - x_6x_3 - (y_2 - f_5)x_5) - \dot{x}_6 \quad (5.17)$$

Note that with the three presented ARR's all the given measurements $y_1 - y_8$ are used.

Before going to the simulation results two more ARR's will be shown, namely two ARR's which do not require to solve the differential equation in c_{10} but together give the same fault detection and isolation results (or even better) as the one just calculated. If one does not necessarily need a minimal amount of ARR's, with FMSO φ_1 the following expression can be obtained,

$$f_4 = \frac{\tau_v}{\nu\sqrt{y_4}} \left(\dot{y}_3 - \frac{\dot{y}_4 y_3}{2y_4} + \frac{y_3}{\tau_v} \right) - A_v^* y_8 \quad (5.18)$$

which is an ARR for only the fault f_4 and it does not require to solve the differential equation. This works because the variable x_3 does not need to be calculated by x_{13} since it can be also taken as the measured value y_3 . Then x_{13} can be determined directly with c_9 . In (5.18) it can be seen that the fault f_4 is not detectable if the system pressure $y_4 = 0$, which is the case when the pump is not running, the membrane is broken and no pressure can be generated anymore.

If the ARR in (5.18) is chosen to detect f_4 , then another one is demanded for isolating f_2 from f_6 , which can be done with φ_{29} containing only f_2 and f_5 . This leads to the same ARR as for φ_{30} , just that also here x_3 is taken from the directly measured value y_3 .

The five calculated ARR's are simulated with the model developed in chapter 3. When implementing the actual residual generators, the part of the ARR with all faults and its respective derivatives needs to be set equal to zero. When the ARR's are deduced it is actually not necessary to consider the faults and its derivatives, but neglecting them in this step, takes the opportunity to later analyze how the output of residual will be for a given fault.

5.2 Simulation of the fault diagnosis system

All the residuals are simulated with the aforementioned inputs when simulating the plant in section 3.7. The faults are not shown here since as much information as possible needs to be interpreted from the outputs of the residuals.

Figure 5.2 displays the outputs of the residuals in the faultless case. The residuals show

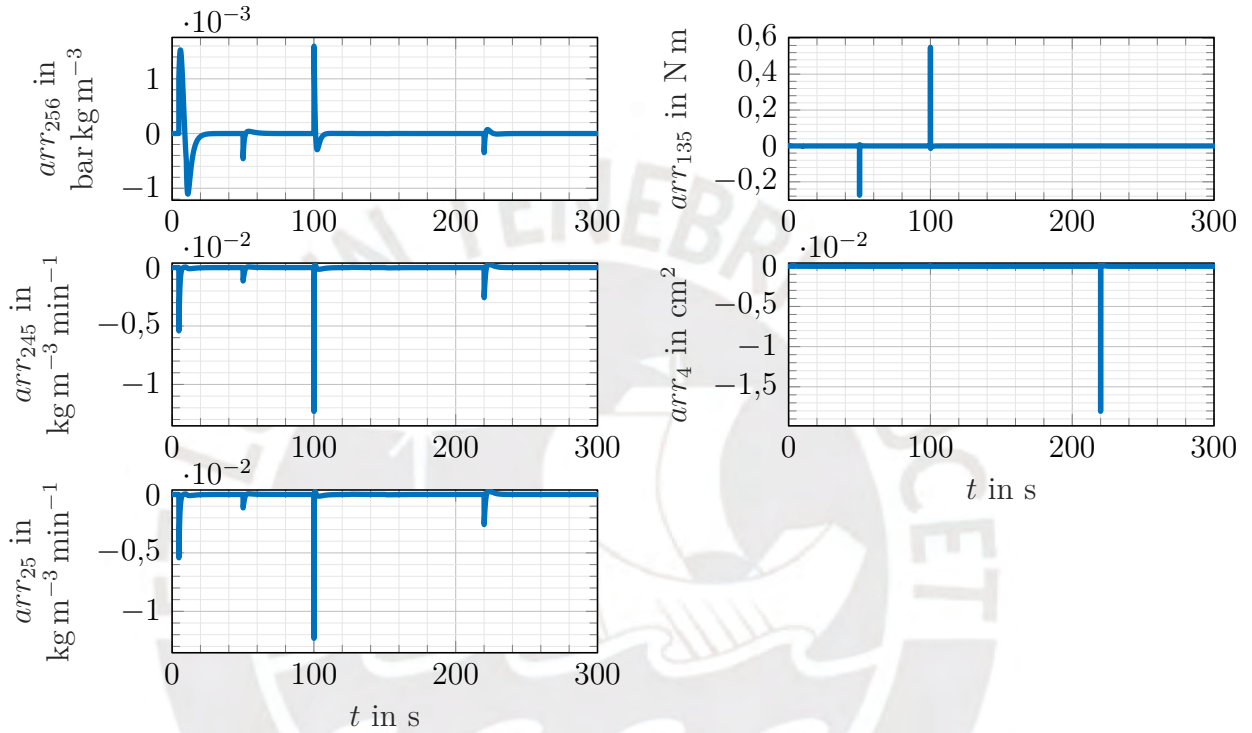


Fig. 5.2: Output of the residual generators in the faultless case

some spikes when the plant changes its set point, even though it should be equal to zero over the entire time period because no fault occurs. This is unavoidable and is caused by the derivatives of the measured variables needed for calculating the residuals. The derivative is calculated in the following way:

$$\dot{y}(t) = \frac{y(t) - y(t - \Delta t)}{\Delta t} \quad (5.19)$$

Unfortunately, equation (5.19) will never be exact because the exact value is represented by [Bro96]:

$$\dot{y}(t) = \lim_{\Delta t \rightarrow 0} \frac{y(t) - y(t - \Delta t)}{\Delta t} \quad (5.20)$$

which is not realizable. Improvements can be accomplished by minimizing Δt , but this can lead to problems when dealing with noise of the sensors. Another way would be filtering the

outputs, which leads to a delay of fault detection when a fault occurs. Since noise is here not considered, Δt is set to a value of 0.001 s and the values of the two inputs y_7 and y_8 are made continuous and differentiable by filtering them with the transfer function shown in equation (5.21).

$$G(s) = \frac{1}{(0.01)^2 s^2 + 2 \cdot 0.01 s + 1} \quad (5.21)$$

This filtering causes a slower change of the system and therefore less deviations when calculating the derivative. This filter was also applied in the simulations in section 3.7 to obtain comparable results. Note that when operating in closed loop the input is determined by the controller. It is therefore recommended to use a controller with a continuous differentiable control signal.

The spikes can also be caused by numerical problems which occur when dealing with huge and small numbers within an ARR. This case arises especially then, when the ARRs contain the derivative of faults which occur in an abrupt way.

The errors in this case are tolerable because it is more important how sensitive a residual is when a fault occurs. If it can be clearly distinguished between a fault and a calculation problem, it is sufficient as seen in the following figures where faults occur.

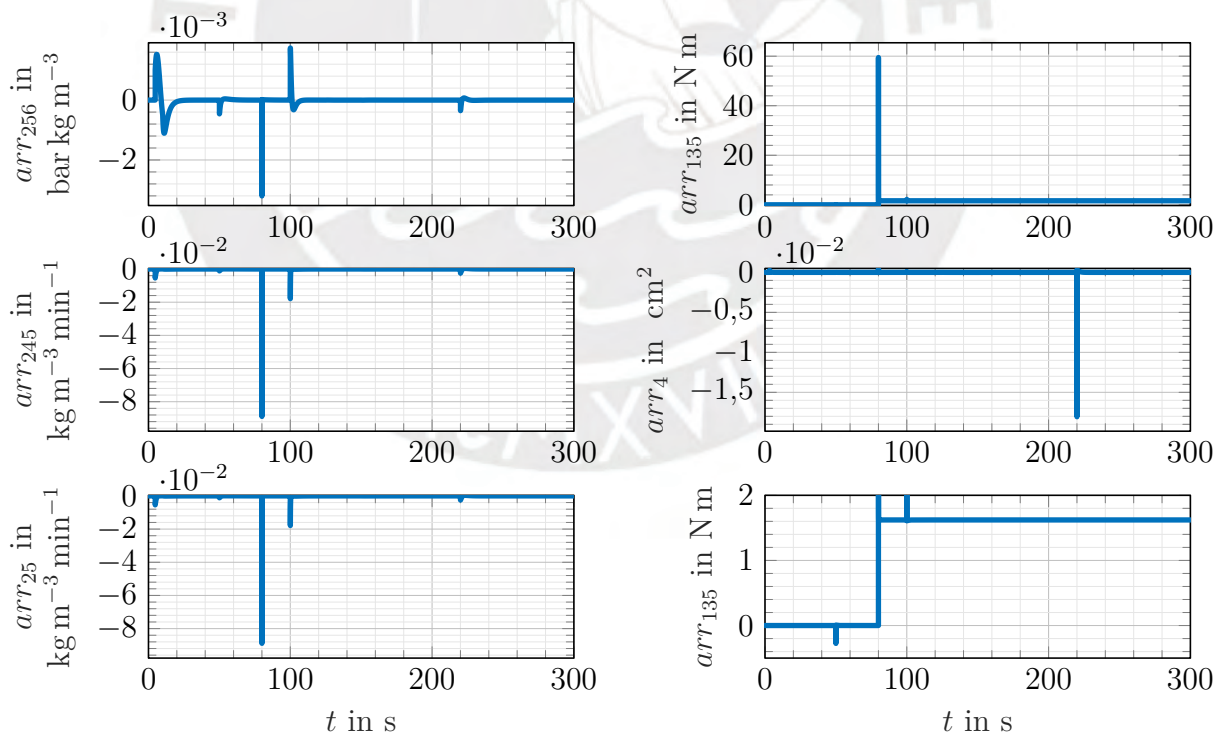


Fig. 5.3: Output of the residual generators when fault f_1 gets active

Figure 5.3 shows the output of the residuals when fault f_1 occurs. Only the residual arr_{135} has a significant change with respect to the faultless case. The fault occurs at 80 s when the residual jumps to a value of 60 N m and immediately back to a value of 1.6 N m where it stays (seen in the figure at the bottom right of 5.3). The output of arr_{135} in equation (5.11) shows that not only the absolute value of f_1 affects the ARR but also its time derivative which generates the peak when the fault occurs. The value of the residual stays constant afterwards, although there are still set point changes of the plant. As fault f_1 in (5.11) is not coupled to any variable which changes over time, it can be said, that the fault occurs as a step. The second peak at 100 s is because of the derivative problem mentioned earlier and just reaches the value 2.0, which can be neglected. As soon as \dot{f}_1 gets zero, the residual should show $\frac{\pi(d+cn_p)}{30V_d}f_1$ so that the value of f_1 is $\frac{1.6 \cdot 30V_d}{\pi(d+cn_p)} \approx 11 \text{ min}^{-1}$ which was the value the fault was simulated with.

Note that fault f_1 was simulated here. When the ARRs are applied to the real system, one cannot say if fault f_1 or fault f_3 is active because they are not isolable from each other, in the sense of structural analysis. However, when analyzing the ARRs some preferences for a certain fault can be made. In this case it can also be assumed that the output is generated by the fault f_3 . f_3 appears in equation (5.11) without its derivative. Therefore, the residual shows the direct value of the fault, in this case even in its unit of N m. This means that the fault first brakes the pump of the motor with 60 N m and afterwards with 1.6 N m. This should be checked if it is plausible for fault f_3 . Moreover, it can also be checked what happens with the measured variables y_x when f_3 has this shape. This is not shown here, but if f_3 occurs in this shape, the other ARRs will be also affected because of the fast change of the rotational speed of the pump and the rapid pressure drop which leads to more errors by the derivatives. Nevertheless, it is just a preference and needs to be handled with care.

Figure 5.4 shows the outputs when fault f_2 occurs which is the membrane fouling. In this fault f_2 can be determined with certainty because for every other fault different residuals would be not equal to zero. Even though the residuals have little deviation from zero, it can be distinguished from the behavior of the faultless case. Unfortunately, none of the ARRs can be used to directly determine the value of f_2 because none of them was brought to the form as in (5.11).

The residuals arr_{245} and arr_{25} should show the same behavior because they are calculated in the same way except that in arr_{25} x_3 is taken directly from its measured value and in arr_{245} x_3 is determined by the solution of the differential equation in c_{10} . In comparison arr_{245} has a little bump at 10 s where arr_{25} does not have one. In this case the initial value of the integrator was set to $20A_v^*$ even though a recommended value would be between zero

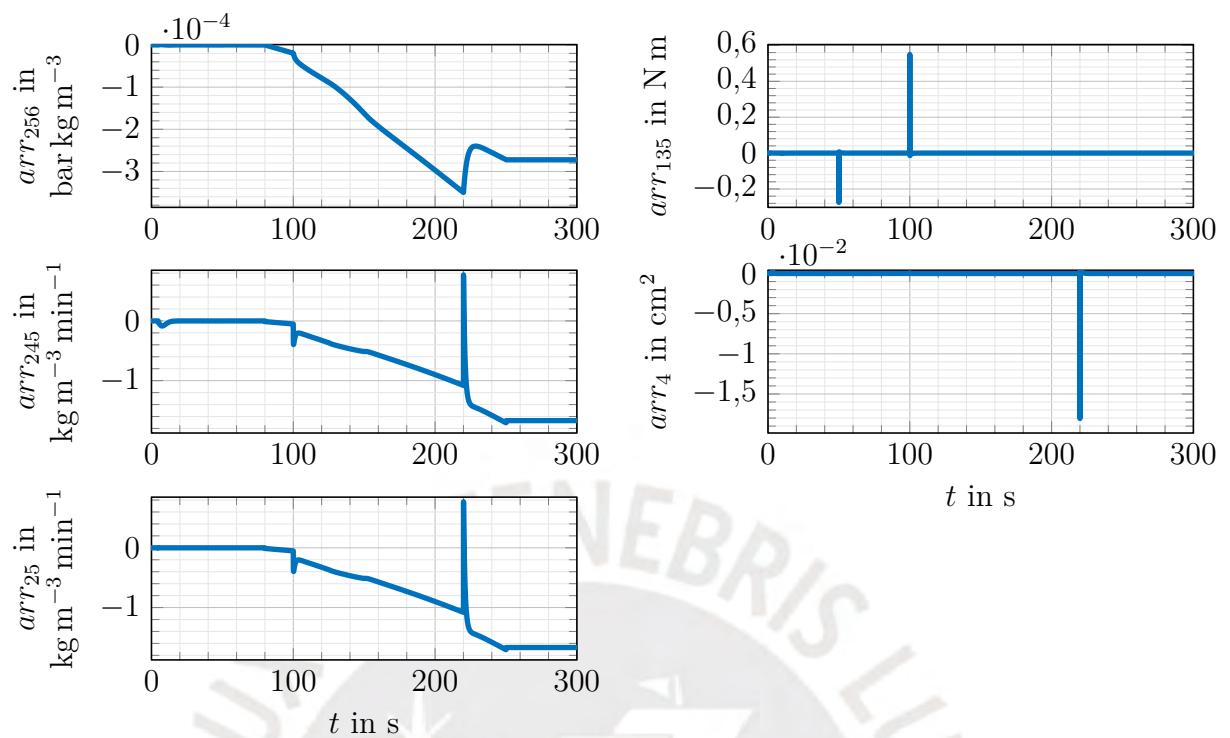
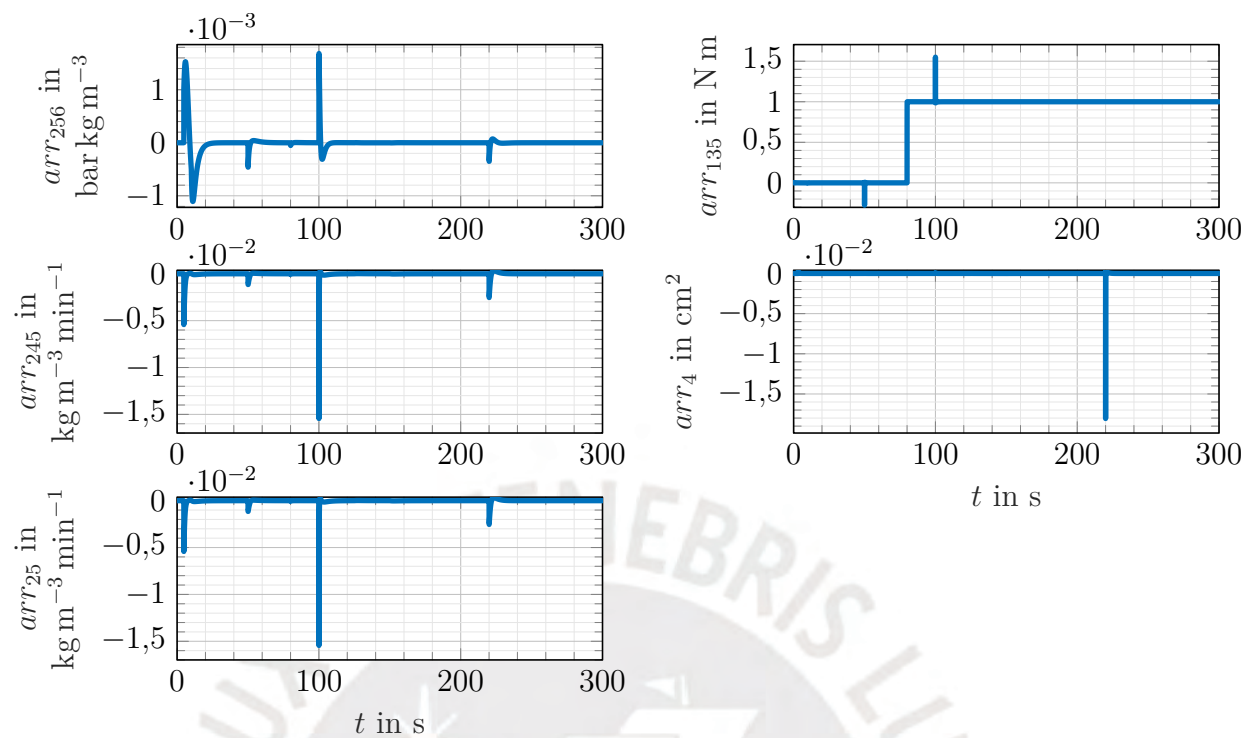
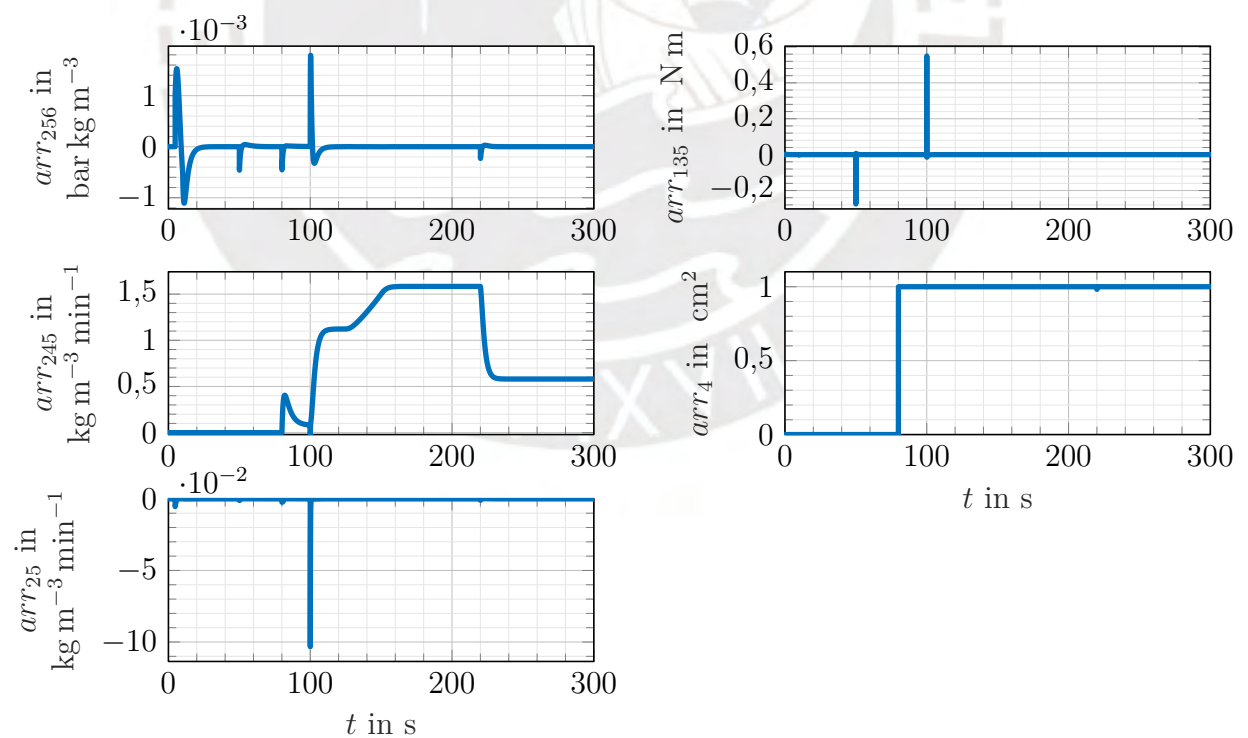


Fig. 5.4: Output of the residual generators when fault f_2 gets active

and A_v^* . The error vanishes after 20 s when the state x_{13} has converged close enough to its real value. Afterwards, both residuals show the same value.

Fault f_3 is detected with the arr_{135} , shown in figure 5.5, which outputs exactly the value of the simulated fault. A step with the height of 1 N m was simulated. Note that in this case the residual does not overshoot as with fault f_1 , which is a good indicator that this might be fault f_3 not f_1 .

Fig. 5.5: Output of the residual generators when fault f_3 gets activeFig. 5.6: Output of the residual generators when fault f_4 gets active

Fault f_4 is reliably detected by the both presented residuals whereas arr_4 exposes directly the value of f_4 but arr_{245} only shows that the fault is active, as shown in figure 5.6. The fault f_4 is simulated with a step with height 1 cm^2 .

Also fault f_5 and f_6 can be clearly detected by the according residuals shown in figure 5.7 and 5.8. Fault f_5 is simulated with a step height of 1 l min^{-1} and f_6 with 0.01 S m^{-1} which corresponds to a measurement error of 0.1 kg m^{-3} for the salt concentration of the permeate stream. The value of fault f_6 is already difficult to detect, because the change of the output of the residual does not differ too much from the values shown before where the according ARR should not show a fault. But at least the peak is ten times higher as usual which would be sufficient to trigger an alert that more attention needs to be payed to this ARR.

The residuals presented in this section show satisfying results. The three selected FMSOs in section 4.5.1 are sufficient to detect and isolate all the possible faults but have the disadvantage that determining the magnitude of the occurring fault is more difficult than with the additional ones. Note that the characteristic of every residual can be changed according to the interests of fault detection and isolation. For example, if the magnitude of fault f_2 is required, the according residual must be calculated in a different way such that the output is directly equal to the value of f_2 . If the magnitude of every fault needs to be shown, a residual could be calculated with different outputs. Also its sensitivity to numerical problems can be affected by changing, for example, the units of the measured values.

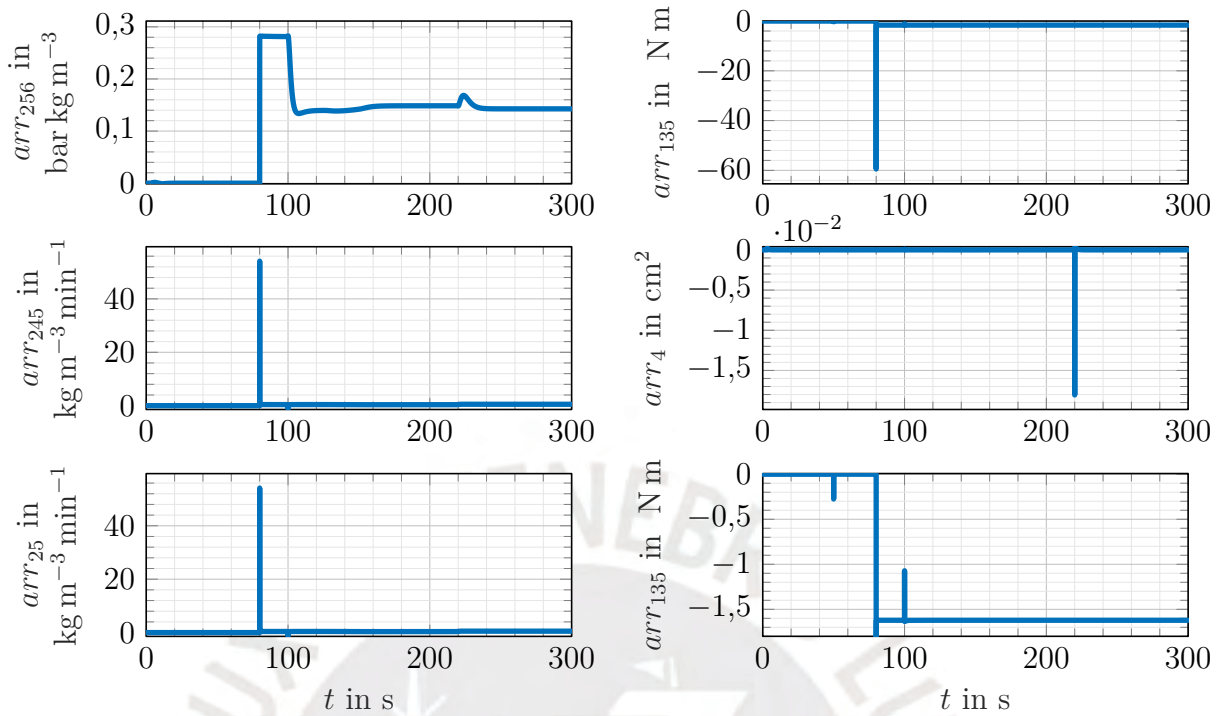


Fig. 5.7: Output of the residual generators when fault f_5 gets active

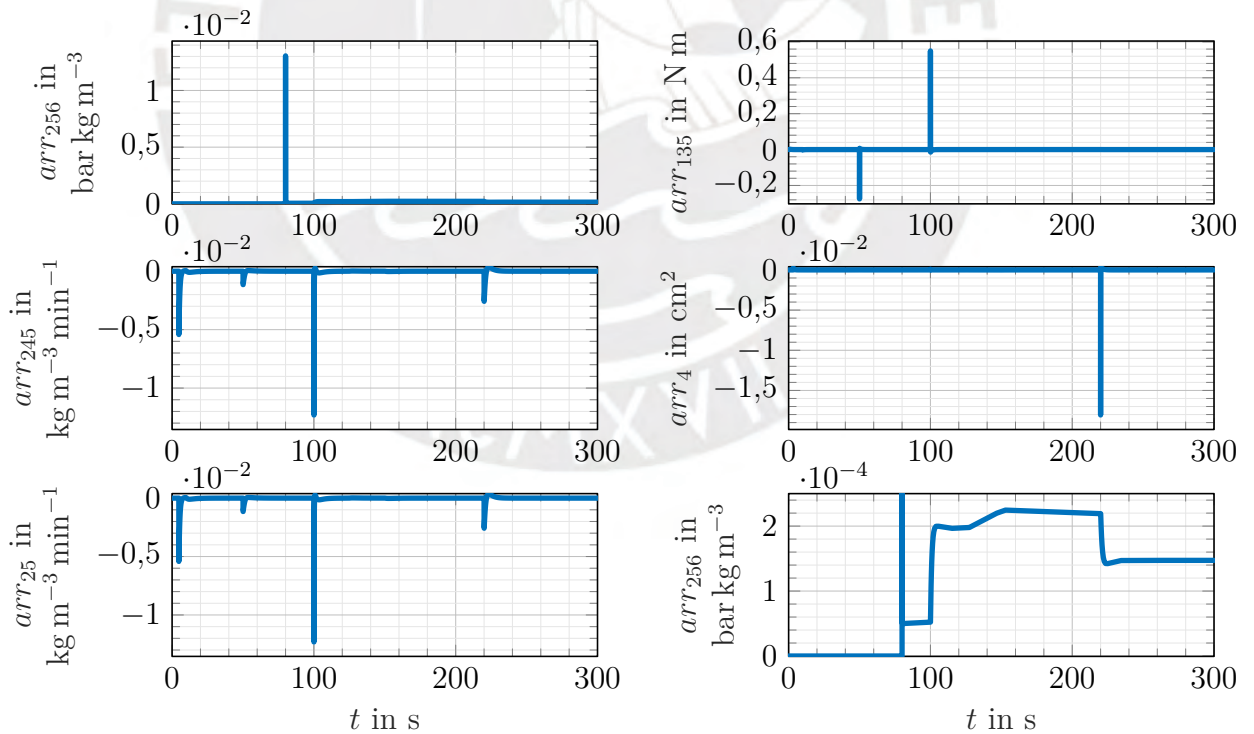


Fig. 5.8: Output of the residual generators when fault f_6 gets active

Conclusion

The results of the presented work show that structural analysis is a powerful tool for fault diagnosis. Faults of a system can be evaluated fast, concerning their detectability and isolability. Compared to the approaches made with, for example, neural networks, structural analysis has the drawback that a mathematical model is required. However, the structural analysis has the advantage that faults can be analyzed in a more deterministic way, and no learning procedure is required.

The mathematical model of the developed reverse osmosis system extends the modeling approaches presented in previous works. Modeling the high pressure pump's interaction with the membrane module and the brine valve results in a more detailed description of the system behavior and the possibility of detecting more faults. Furthermore, the model provides a good base for other works that require a mathematical model. The entire reverse osmosis plant's behavior taken for structural analysis has been modeled with 20 equations that contain 16 variables. Six variables are measured directly with sensors, and two inputs are said to be known. The differential algebraic model contains four differential equations and is non-linear.

Six faults have been considered for fault-diagnosis: Two sensor faults, two actuator faults and two plant faults. With the graph based structural analysis, 73 fault-driven minimal

structurally overdetermined sets have been found. A FMSO set is an overdetermined set of equations that can be used for fault diagnosis because an overdetermined system of equations has either one or no solution.

The six faults considered are all detectable, but two of them are not isolable from each other. Three of the 73 FMSOs have been selected. Considering that only one fault at a time occurs, three FMSOs are then a minimum number to consider for the detection and isolation of six faults.

A structural model does not consider any time dependence, non-linearity, or the variables' actual values. For implementing the fault diagnosis system, it is necessary to deduce the analytical redundancy relations, which are the causal interpretation of the FMSO sets determined via structural analysis.

Despite neglecting the differential and non-linear constraints that can lead to problems when deducing the ARR, the three selected FMSOs could be used to deduce the necessary residuals for fault detection. Two additional ARRs have been calculated, which could also be used for fault detection and have the advantage that the magnitude of at least one more fault can be determined.

For the residual generators extracted from the ARRs, the differentiation of a measured value is required. Unfortunately, a perfect differentiation is not realizable. Therefore when implementing the residual generators, further considerations according to the differentiation and integration, noise, numerical errors must be made.

When the reverse osmosis system is operated with a controller, the differentiation problem can be minimized by choosing a controller with a continuous and differentiable output that prevents steps and fast dynamics. Thus, fewer deviations occur when calculating the differentiation of the values.

The next step is comparing the model with the real system, which also requires determining the system parameters. Afterwards, the residuals can be implemented and evaluated. Furthermore, it can be considered adding a sensor for measuring the rotational speed of the pump as then the two indistinguishable faults will become isolable. However, it is shown that a preference between the two faults can be made by analyzing the residuals.

Fault diagnosis always means dealing with something that is not known but would be useful if it were. When modeling the faults, a trade-off is necessary because not all faults that can occur can be modeled and evaluated. Nevertheless, with the presented fault diagnosis system, it is possible to get alarmed as soon as the plant's behavior is not consistent anymore. Based on this alarm, further measures can be performed to prevent more damage or even a system's failure.

Bibliography

- [And02] ANDERSON, Ian: *A First Course in Discrete Mathematics*. Springer, 2002. ISBN: 978-1-4612-6879-6
- [ABE⁺09] ARMENGOL, J. ; BREGON, A. ; ESCOBET, T. ; GELSO, E. ; KRYSANDER, M. ; NYBERG, M. ; OLIVE, X. ; PULIDO, B. ; TRAVÉ-MASSUYÈS, L.: Minimal Structurally Overdetermined Sets for Residual Generation: A Comparison of Alternative Approaches. In: *IFAC Proceedings Volumes* 42 (2009) No. 8, pp. 1480–1485. DOI: 10.3182/20090630-4-ES-2003.00241
- [BCA⁺15] BAAWAIN, Mahad ; CHOUDRI, BS ; AHMED, Mushtaque ; PURNAMA, Anton: *Recent Progress in Desalination, Environmental and Marine Outfall Systems*. Springer, 2015. ISBN: 978-3-319-19122-5
- [Bak00] BAKER, Richard W.: *Membrane Technology and Applications*. McGraw-Hill Professional Engineering. McGraw-Hill, 2000. ISBN: 978-0-07-135440-0
- [BCC09] BARTMAN, Alex R. ; CHRISTOFIDES, Panagiotis D. ; COHEN, Yoram: Nonlinear Model-Based Control of an Experimental Reverse-Osmosis Water Desalination System. In: *Industrial & Engineering Chemistry Research* 48 (July 2009) No. 13, pp. 6126–6136. DOI: 10.1021/ie900322x
- [BKL⁺16] BLANKE, Mogens ; KINNAERT, Michel ; LUNZE, Jan ; STAROSWIECKI, Marcel: *Diagnosis and Fault-Tolerant Control*. Springer, 2016. ISBN: 978-3-662-47942-1
- [Bro96] BROWDER, Andrew: *Mathematical Analysis*. Red. by AXLER, S. ; GEHRING, F. W. ; RIBET, K. A. Springer, 1996. ISBN: 978-1-4612-6879-6

- [Bun86] BUNGAY, Janet K.: *Synthetic Membranes: Science, Engineering and Applications*. Springer, 1986. ISBN: 978-94-010-8596-0
- [CCE06] CATH, T ; CHILDRESS, A ; ELIMELECH, M: Forward Osmosis: Principles, Applications, and Recent Developments. In: *Journal of Membrane Science* 281 (Sept. 15, 2006) No. 1-2, pp. 70–87. DOI: 10.1016/j.memsci.2006.05.048
- [Con15] CONFALONIERI, Sara: *The Unattainable Attempt to Avoid the Casus Irreducibilis for Cubic Equations*. Springer, 2015. ISBN: 978-3-658-09274-0
- [DW19] DEBELE NEGEWO, Bekele ; WARD, Christopher S: *The Role of Desalination in an Increasingly Water-Scarce World*. The World Bank, 2019
- [EE02] EL-DESSOUKY, H. T. ; ETTOUNEY, H. M.: *Fundamentals of Salt Water Desalination*. Elsevier, 2002. ISBN: 0-444-50810-4
- [DM58] DULMAGE, A. L. ; MENDELSON, N. S.: Coverings of Bipartite Graphs. In: *Canadian Journal of Mathematics* 10 (1958), pp. 517–534. DOI: 10.4153/CJM-1958-052-0
- [EBP⁺19] ESCOBET, Teresa ; BREGON, Anibal ; PULIDO, Belarmino ; PUIG, Vicenç (eds.): *Fault Diagnosis of Dynamic Systems: Quantitative and Qualitative Approaches*. Springer, 2019. ISBN: 978-3-030-17727-0
- [FC17] FIGOLI, Alberto ; CRISCUOLI, Alessandra (eds.): *Sustainable Membrane Technology for Water and Wastewater Treatment*. Springer, 2017. ISBN: 978-981-10-5621-5
- [FKÅ17] FRISK, Erik ; KRYSANDER, Mattias ; ÅSLUND, Jan: Analysis and Design of Diagnosis Systems Based on the Structural Differential Index. In: *IFAC-PapersOnLine* 50 (July 2017) No. 1, pp. 12236–12242. DOI: 10.1016/j.ifacol.2017.08.2129
- [FKJ17] FRISK, Erik ; KRYSANDER, Mattias ; JUNG, Daniel: A Toolbox for Analysis and Design of Model Based Diagnosis Systems for Large Scale Models. In: *IFAC-PapersOnLine* 50 (July 2017) No. 1, pp. 3287–3293. DOI: 10.1016/j.ifacol.2017.08.504

- [GKB07] GAMBIER, A. ; KRASNIK, A. ; BADREDDIN, E.: Dynamic Modeling of a Simple Reverse Osmosis Desalination Plant for Advanced Control Purposes. In: *American Control Conference*. July 2007. DOI: 10.1109/acc.2007.4283019
- [GFP11] GARCIA-ALVAREZ, D. ; FUENTE, M.J. ; PALACIN, L.G.: Monitoring and Fault Detection in a Reverse Osmosis Plant Using Principal Component Analysis. In: *IEEE Conference on Decision and Control*. Dec. 2011, pp. 3044–3049. DOI: 10.1109/CDC.2011.6160345
- [Hal35] HALL, P: ON REPRESENTATIVES OF SUBSETS. In: (1935)
- [Ise08] ISERMANN, R.: *Mechatronische Systeme Grundlagen*. Springer, 2008. ISBN: 978-3-540-32336-5
- [Ise06] ISERMANN, Rolf: *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer, 2006. ISBN: 978-3-540-24112-6
- [Kim17] KIM, Albert S.: Review of Basics Reverse Osmosis Process Modeling: A New Combined Fouling Index Proposed. In: *Membrane Journal* 27 (Aug. 2017) No. 4, pp. 291–312. DOI: 10.14579/membrane_journal.2017.27.4.291
- [KH05] KIM, Suhan ; HOEK, Eric M. V.: Modeling Concentration Polarization in Reverse Osmosis Processes. In: *Desalination* 186 (Dec. 2005) No. 1-3, pp. 111–128. DOI: 10.1016/j.desal.2005.05.017
- [Kry06] KRYSANDER, Mattias: *Design and Analysis of Diagnosis Systems Using Structural Methods*. 2006
- [KÅF10] KRYSANDER, Mattias ; ÅSLUND, Jan ; FRISK, Erik: A Structural Algorithm for Finding Testable Sub-Models and Multiple Fault Isolability Analysis. In: (2010)
- [KÅN05] KRYSANDER, Mattias ; ÅSLUND, Jan ; NYBERG, Mattias: An Efficient Algorithm for Finding Over-Constrained Sub-Systems for Construction of Diagnostic Tests. In: (2005)
- [Kuc15] KUCERA, Jane: *Reverse Osmosis: Industrial Processes and Applications*. 2nd ed. John Wiley & Sons, 2015. ISBN: 978-1-118-63974-0
- [Lev19] LEVIN, Oscar: *Discrete Mathematics: An Open Introduction*. 3rd ed. Dec. 29, 2019. ISBN: 978-1-79290-169-0

- [MBC⁺08] MCFALL, Charles W. ; BARTMAN, Alex ; CHRISTOFIDES, Panagiotis D. ; COHEN, Yoram: Control and Monitoring of a High Recovery Reverse Osmosis Desalination Process. In: *Industrial & Engineering Chemistry Research* 47 (Sept. 3, 2008) No. 17, pp. 6698–6710. DOI: 10.1021/ie071559b
- [Naj16] NAJAFI, Fazil T: Environmental Impact Cost Analysis of Multi-Stage Flash, Multi-Effect Distillation, Mechanical Vapor Compression, and Reverse Osmosis Medium-Size Desalination Facilities. In: *ASCE's 123rd Annual Conference and Exposition*. 2016
- [PGB⁺14] PASCUAL, Xavier ; GU, Han ; BARTMAN, Alex ; ZHU, Aihua ; RAHARDIANTO, Anditya ; GIRALT, Jaume ; RALLO, Robert ; CHRISTOFIDES, Panagiotis D. ; COHEN, Yoram: Fault Detection and Isolation in a Spiral-Wound Reverse Osmosis (RO) Desalination Plant. In: *Industrial & Engineering Chemistry Research* 53 (Feb. 2014) No. 8, pp. 3257–3271. DOI: 10.1021/ie403603x
- [PTC⁺15] PÉREZ, C G ; TRAVÉ-MASSUYÈS, L ; CHANTHERY, E ; SOTOMAYOR, J: Decentralized Diagnosis in a Spacecraft Attitude Determination and Control System. In: *Journal of Physics: Conference Series* 659 (2015). DOI: 10.1088/1742-6596/659/1/012054
- [Pér17] PÉREZ ZUÑIGA, Carlos Gustavo: *Structural Analysis for the Diagnosis of Distributed Systems*. 2017
- [PCT⁺17] PÉREZ-ZUÑIGA, C.G. ; CHANTHERY, E. ; TRAVÉ-MASSUYÈS, L. ; SOTOMAYOR, J.: Fault-Driven Structural Diagnosis Approach in a Distributed Context. In: *IFAC-PapersOnLine* 50 (2017) No. 1, pp. 14254–14259. DOI: 10.1016/j.ifacol.2017.08.1819
- [PCT⁺18] PÉREZ-ZUÑIGA, C.G. ; CHANTHERY, E. ; TRAVÉ-MASSUYÈS, L. ; SOTOMAYOR, J. ; ARTIGUES, C.: Decentralized Diagnosis via Structural Analysis and Integer Programming. In: *IFAC-PapersOnLine* 51 (2018) No. 24, pp. 168–175. DOI: 10.1016/j.ifacol.2018.09.551
- [PF90] POTHEN, Alex ; FAN, Chin-Ju: Computing the Block Triangular Form of a Sparse Matrix. In: *ACM Transactions on Mathematical Software* 16 (Dec. 1990) No. 4, pp. 303–324. DOI: 10.1145/98267.98287
- [Pot19] POTTMEYER, Lukas: *Diskrete Mathematik: Ein Kompakter Einstieg*. Springer, 2019. ISBN: 978-3-662-59662-3

- [RKS⁺13] RATHORE, N. S. ; KUNDARIYA, N. ; SADISTAP, S. ; NARAIN, A.: Mathematical Modeling and Simulation of Concentration Polarization Layer in Reverse Osmosis Process. In: *Students Conference on Engineering and Systems*. 2013, pp. 1–4. DOI: 10.1109/SCES.2013.6547547
- [RSP⁺19] RIVAS-PEREZ, Raul ; SOTOMAYOR-MORIANO, Javier ; PÉREZ-ZUÑIGA, Gustavo ; SOTO-ANGLES, Mario E.: Real-Time Implementation of an Expert Model Predictive Controller in a Pilot-Scale Reverse Osmosis Plant for Brackish and Seawater Desalination. In: *Applied Sciences* 9 (July 2019) No. 14, pp. 2932–2948. DOI: 10.3390/app9142932
- [SCS11] SALLAMI, Abderrahmene ; CHAABENE, Abderrahmene BEN ; SALLAMI, Anis: Robust Fault Diagnosis of a Reverse Osmosis Desalination System Modeled by Bond Graph Approach. In: *International Journal of Computer Science and Network Security* 11 (2011) No. 12, pp. 105–111
- [SR14] SHATAT, Mahmoud ; RIFFAT, Saffa B: Water Desalination Technologies Utilizing Conventional and Renewable Energy Sources. In: *International Journal of Low-Carbon Technologies* 9 (2014) No. 1, pp. 1–19. DOI: 10.1093/ijlct/cts025
- [Sol] *Solving Cubic Equations*. University of Melbourne. URL: <http://www2.trinity.unimelb.edu.au/~rbroekst/MathX/Cubic%20Formula.pdf> (Date: 11/18/2020)
- [Sot16] SOTO ANGLAS, Mario Eduardo: *Diseño de Un Sistema de Detección y Diagnóstico de Fallas Basado En Modelo Para Una Planta Desalinizadora de Agua de Mar*. Master's thesis. Pontificia Universidad Católica del Perú, Escuela de posgrado, 2016
- [SK66] SPIEGLER, K. S. ; KEDEM, O.: Thermodynamics of Hyperfiltration (Reverse Osmosis): Criteria for Efficient Membranes. In: *Desalination* 1 (Dec. 1, 1966) No. 4, pp. 311–326. DOI: 10.1016/S0011-9164(00)80018-1
- [STP⁺08] SYAFIIE, S. ; TADEO, F. ; PALACIN, L. ; de PRADA, C. ; SALAZAR, J.: Modelling for Dynamic Simulation of Pretreatment in Reverse Osmosis Plants. In: *IEEE International Conference on Industrial Engineering and Engineering Management*. 2008, pp. 1663–1667. DOI: 10.1109/ieem.2008.4738155
- [TSB⁺18] THIMMARAJU, M. ; SREEPADA, D. ; BABU, G. S. ; DASARI, B. K. ; VELPULA, S. K. ; VALLEPU, N.: Desalination of Water. In: *Desalination and Water Treatment*. InTech, Sept. 2018. DOI: 10.5772/intechopen.78659

- [WCH⁺11] WANG, Lawrence K. ; CHEN, Jiaping Paul ; HUNG, Yung-Tse ; SHAMMAS, Nazih K. (eds.): *Membrane and Desalination Technologies*. Springer, 2011. ISBN: 978-1-58829-940-6
- [Wat17] WATTER, H.: *Hydraulik Und Pneumatik*. Springer, 2017. ISBN: 978-2-658-18554-1



APPENDIX A

All found FMSOs

φ	\mathcal{F}_φ
1. $\{c_9, c_{10}, c_{15}, c_{16}, c_{20}\}$	$\{f_4\}$
2. $\{c_3, c_4, c_5, c_{11}, c_{13}, c_{14}, c_{16}, c_{17}\}$	$\{f_2, f_5, f_6\}$
3. $\{c_3, c_4, c_5, c_9, c_{10}, c_{11}, c_{13}, c_{14}, c_{15}, c_{17}, c_{20}\}$	$\{f_2, f_4, f_5, f_6\}$
4. $\{c_2, c_4, c_5, c_6, c_7, c_8, c_{12}, c_{13}, c_{14}, c_{15}, c_{16}, c_{18}, c_{19}\}$	$\{f_1, f_2, f_3, f_5\}$
5. $\{c_2, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{12}, c_{13}, c_{14}, c_{16}, c_{18}, c_{19}, c_{20}\}$	$\{f_1, f_2, f_3, f_4, f_5\}$
6. $\{c_2, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{12}, c_{13}, c_{14}, c_{15}, c_{18}, c_{19}, c_{20}\}$	$\{f_1, f_2, f_3, f_4, f_5\}$
7. $\{c_2, c_3, c_5, c_6, c_7, c_8, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}, c_{16}, c_{17}, c_{18}, c_{19}\}$	$\{f_1, f_3, f_5, f_6\}$
8. $\{c_2, c_3, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}, c_{16}, c_{17}, c_{18}, c_{19}, c_{20}\}$	$\{f_1, f_3, f_4, f_5, f_6\}$
9. $\{c_2, c_3, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}, c_{17}, c_{18}, c_{19}, c_{20}\}$	$\{f_1, f_3, f_4, f_5, f_6\}$
10. $\{c_2, c_3, c_4, c_6, c_7, c_8, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}, c_{16}, c_{17}, c_{18}, c_{19}\}$	$\{f_1, f_2, f_3, f_5, f_6\}$
11. $\{c_2, c_3, c_4, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}, c_{16}, c_{17}, c_{18}, c_{19}, c_{20}\}$	$\{f_1, f_2, f_3, f_4, f_5, f_6\}$
12. $\{c_2, c_3, c_4, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}, c_{17}, c_{18}, c_{19}, c_{20}\}$	$\{f_1, f_2, f_3, f_4, f_5, f_6\}$
13. $\{c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_{11}, c_{12}, c_{14}, c_{15}, c_{16}, c_{17}, c_{18}, c_{19}\}$	$\{f_1, f_2, f_3, f_5, f_6\}$
14. $\{c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_{11}, c_{12}, c_{13}, c_{15}, c_{16}, c_{17}, c_{18}, c_{19}\}$	$\{f_1, f_2, f_3, f_6\}$
15. $\{c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}, c_{17}, c_{18}, c_{19}\}$	$\{f_1, f_2, f_3, f_5, f_6\}$
16. $\{c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{14}, c_{16}, c_{17}, c_{18}, c_{19}, c_{20}\}$	$\{f_1, f_2, f_3, f_4, f_5, f_6\}$
17. $\{c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{14}, c_{15}, c_{17}, c_{18}, c_{19}, c_{20}\}$	$\{f_1, f_2, f_3, f_4, f_5, f_6\}$
18. $\{c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{16}, c_{17}, c_{18}, c_{19}, c_{20}\}$	$\{f_1, f_2, f_3, f_4, f_6\}$

φ	\mathcal{F}_φ
19. $\{c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{15}, c_{17}, c_{18}, c_{19}, c_{20}\}$	$\{f_1, f_2, f_3, f_4, f_6\}$
20. $\{c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}, c_{17}, c_{18}, c_{19}, c_{20}\}$	$\{f_1, f_2, f_3, f_4, f_5, f_6\}$
21. $\{c_1, c_7, c_8, c_{14}, c_{15}, c_{16}, c_{19}\}$	$\{f_1, f_3, f_5\}$
22. $\{c_1, c_7, c_8, c_9, c_{10}, c_{14}, c_{16}, c_{19}, c_{20}\}$	$\{f_1, f_3, f_4, f_5\}$
23. $\{c_1, c_7, c_8, c_9, c_{10}, c_{14}, c_{15}, c_{19}, c_{20}\}$	$\{f_1, f_3, f_4, f_5\}$
24. $\{c_1, c_3, c_4, c_5, c_7, c_8, c_{11}, c_{13}, c_{15}, c_{16}, c_{17}, c_{19}\}$	$\{f_1, f_2, f_3, f_6\}$
25. $\{c_1, c_3, c_4, c_5, c_7, c_8, c_{11}, c_{13}, c_{14}, c_{15}, c_{17}, c_{19}\}$	$\{f_1, f_2, f_3, f_5, f_6\}$
26. $\{c_1, c_3, c_4, c_5, c_7, c_8, c_9, c_{10}, c_{11}, c_{13}, c_{16}, c_{17}, c_{19}, c_{20}\}$	$\{f_1, f_2, f_3, f_4, f_6\}$
27. $\{c_1, c_3, c_4, c_5, c_7, c_8, c_9, c_{10}, c_{11}, c_{13}, c_{15}, c_{17}, c_{19}, c_{20}\}$	$\{f_1, f_2, f_3, f_4, f_6\}$
28. $\{c_1, c_3, c_4, c_5, c_7, c_8, c_9, c_{10}, c_{11}, c_{13}, c_{14}, c_{17}, c_{19}, c_{20}\}$	$\{f_1, f_2, f_3, f_4, f_5, f_6\}$
29. $\{c_1, c_2, c_4, c_5, c_6, c_{12}, c_{13}, c_{14}, c_{15}, c_{16}, c_{18}\}$	$\{f_2, f_5\}$
30. $\{c_1, c_2, c_4, c_5, c_6, c_9, c_{10}, c_{12}, c_{13}, c_{14}, c_{16}, c_{18}, c_{20}\}$	$\{f_2, f_4, f_5\}$
31. $\{c_1, c_2, c_4, c_5, c_6, c_9, c_{10}, c_{12}, c_{13}, c_{14}, c_{15}, c_{18}, c_{20}\}$	$\{f_2, f_4, f_5\}$
32. $\{c_1, c_2, c_4, c_5, c_6, c_7, c_8, c_{12}, c_{13}, c_{15}, c_{16}, c_{18}, c_{19}\}$	$\{f_1, f_2, f_3\}$
33. $\{c_1, c_2, c_4, c_5, c_6, c_7, c_8, c_{12}, c_{13}, c_{14}, c_{16}, c_{18}, c_{19}\}$	$\{f_1, f_2, f_3, f_5\}$
34. $\{c_1, c_2, c_4, c_5, c_6, c_7, c_8, c_{12}, c_{13}, c_{14}, c_{15}, c_{18}, c_{19}\}$	$\{f_1, f_2, f_3, f_5\}$
35. $\{c_1, c_2, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{12}, c_{13}, c_{16}, c_{18}, c_{19}, c_{20}\}$	$\{f_1, f_2, f_3, f_4\}$
36. $\{c_1, c_2, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{12}, c_{13}, c_{15}, c_{18}, c_{19}, c_{20}\}$	$\{f_1, f_2, f_3, f_4\}$
37. $\{c_1, c_2, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{12}, c_{13}, c_{14}, c_{18}, c_{19}, c_{20}\}$	$\{f_1, f_2, f_3, f_4, f_5\}$
38. $\{c_1, c_2, c_3, c_5, c_6, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}, c_{16}, c_{17}, c_{18}\}$	$\{f_5, f_6\}$
39. $\{c_1, c_2, c_3, c_5, c_6, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}, c_{16}, c_{17}, c_{18}, c_{20}\}$	$\{f_4, f_5, f_6\}$
40. $\{c_1, c_2, c_3, c_5, c_6, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}, c_{17}, c_{18}, c_{20}\}$	$\{f_4, f_5, f_6\}$
41. $\{c_1, c_2, c_3, c_5, c_6, c_7, c_8, c_{11}, c_{12}, c_{13}, c_{15}, c_{16}, c_{17}, c_{18}, c_{19}\}$	$\{f_1, f_3, f_6\}$
42. $\{c_1, c_2, c_3, c_5, c_6, c_7, c_8, c_{11}, c_{12}, c_{13}, c_{14}, c_{16}, c_{17}, c_{18}, c_{19}\}$	$\{f_1, f_3, f_5, f_6\}$
43. $\{c_1, c_2, c_3, c_5, c_6, c_7, c_8, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}, c_{17}, c_{18}, c_{19}\}$	$\{f_1, f_3, f_5, f_6\}$
44. $\{c_1, c_2, c_3, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{16}, c_{17}, c_{18}, c_{19}, c_{20}\}$	$\{f_1, f_3, f_4, f_6\}$
45. $\{c_1, c_2, c_3, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{15}, c_{17}, c_{18}, c_{19}, c_{20}\}$	$\{f_1, f_3, f_4, f_6\}$
46. $\{c_1, c_2, c_3, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}, c_{17}, c_{18}, c_{19}, c_{20}\}$	$\{f_1, f_3, f_4, f_5, f_6\}$
47. $\{c_1, c_2, c_3, c_4, c_6, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}, c_{16}, c_{17}, c_{18}\}$	$\{f_2, f_5, f_6\}$
48. $\{c_1, c_2, c_3, c_4, c_6, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}, c_{16}, c_{17}, c_{18}, c_{20}\}$	$\{f_2, f_4, f_5, f_6\}$
49. $\{c_1, c_2, c_3, c_4, c_6, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}, c_{17}, c_{18}, c_{20}\}$	$\{f_2, f_4, f_5, f_6\}$
50. $\{c_1, c_2, c_3, c_4, c_6, c_7, c_8, c_{11}, c_{12}, c_{13}, c_{15}, c_{16}, c_{17}, c_{18}, c_{19}\}$	$\{f_1, f_2, f_3, f_6\}$
51. $\{c_1, c_2, c_3, c_4, c_6, c_7, c_8, c_{11}, c_{12}, c_{13}, c_{14}, c_{16}, c_{17}, c_{18}, c_{19}\}$	$\{f_1, f_2, f_3, f_5, f_6\}$
52. $\{c_1, c_2, c_3, c_4, c_6, c_7, c_8, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}, c_{17}, c_{18}, c_{19}\}$	$\{f_1, f_2, f_3, f_5, f_6\}$
53. $\{c_1, c_2, c_3, c_4, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{16}, c_{17}, c_{18}, c_{19}, c_{20}\}$	$\{f_1, f_2, f_3, f_4, f_6\}$

φ	\mathcal{F}_φ
54. $\{c_1, c_2, c_3, c_4, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{15}, c_{17}, c_{18}, c_{19}, c_{20}\}$	$\{f_1, f_2, f_3, f_4, f_6\}$
55. $\{c_1, c_2, c_3, c_4, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}, c_{17}, c_{18}, c_{19}, c_{20}\}$	$\{f_1, f_2, f_3, f_4, f_5, f_6\}$
56. $\{c_1, c_2, c_3, c_4, c_5, c_6, c_{11}, c_{12}, c_{14}, c_{15}, c_{16}, c_{17}, c_{18}\}$	$\{f_2, f_5, f_6\}$
57. $\{c_1, c_2, c_3, c_4, c_5, c_6, c_{11}, c_{12}, c_{13}, c_{15}, c_{16}, c_{17}, c_{18}\}$	$\{f_2, f_6\}$
58. $\{c_1, c_2, c_3, c_4, c_5, c_6, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}, c_{17}, c_{18}\}$	$\{f_2, f_5, f_6\}$
59. $\{c_1, c_2, c_3, c_4, c_5, c_6, c_9, c_{10}, c_{11}, c_{12}, c_{14}, c_{16}, c_{17}, c_{18}, c_{20}\}$	$\{f_2, f_4, f_5, f_6\}$
60. $\{c_1, c_2, c_3, c_4, c_5, c_6, c_9, c_{10}, c_{11}, c_{12}, c_{14}, c_{15}, c_{17}, c_{18}, c_{20}\}$	$\{f_2, f_4, f_5, f_6\}$
61. $\{c_1, c_2, c_3, c_4, c_5, c_6, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{16}, c_{17}, c_{18}, c_{20}\}$	$\{f_2, f_4, f_6\}$
62. $\{c_1, c_2, c_3, c_4, c_5, c_6, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{15}, c_{17}, c_{18}, c_{20}\}$	$\{f_2, f_4, f_6\}$
63. $\{c_1, c_2, c_3, c_4, c_5, c_6, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}, c_{17}, c_{18}, c_{20}\}$	$\{f_2, f_4, f_5, f_6\}$
64. $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_{11}, c_{12}, c_{15}, c_{16}, c_{17}, c_{18}, c_{19}\}$	$\{f_1, f_2, f_3, f_6\}$
65. $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_{11}, c_{12}, c_{14}, c_{16}, c_{17}, c_{18}, c_{19}\}$	$\{f_1, f_2, f_3, f_5, f_6\}$
66. $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_{11}, c_{12}, c_{14}, c_{15}, c_{17}, c_{18}, c_{19}\}$	$\{f_1, f_2, f_3, f_5, f_6\}$
67. $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_{11}, c_{12}, c_{13}, c_{16}, c_{17}, c_{18}, c_{19}\}$	$\{f_1, f_2, f_3, f_6\}$
68. $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_{11}, c_{12}, c_{13}, c_{15}, c_{17}, c_{18}, c_{19}\}$	$\{f_1, f_2, f_3, f_6\}$
69. $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_{11}, c_{12}, c_{13}, c_{14}, c_{17}, c_{18}, c_{19}\}$	$\{f_1, f_2, f_3, f_5, f_6\}$
70. $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{16}, c_{17}, c_{18}, c_{19}, c_{20}\}$	$\{f_1, f_2, f_3, f_4, f_6\}$
71. $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{15}, c_{17}, c_{18}, c_{19}, c_{20}\}$	$\{f_1, f_2, f_3, f_4, f_6\}$
72. $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{14}, c_{17}, c_{18}, c_{19}, c_{20}\}$	$\{f_1, f_2, f_3, f_4, f_5, f_6\}$
73. $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{17}, c_{18}, c_{19}, c_{20}\}$	$\{f_1, f_2, f_3, f_4, f_6\}$