Model based diagnostic methods with energetic applications

DOI:10.18136/PE.2020.754

PhD Thesis
Author: Anna Ibolya Pózna

Supervisors: Katalin Hangos, DSc
Miklós Gerzson, CSc

University of Pannonia
Faculty of Information Technology
Doctoral School of Information Science

Veszprém
2020

Model based diagnostic methods with energetic applications

Értekezés doktori (PhD) fokozat elnyerése érdekében

Írta:
Pózna Anna Ibolya

Készült a Pannon Egyetem Informatikai Tudományok Doktori Iskolájának keretében

Témavezetők:

Dr. Hangos Katalin
Elfogadásra javaslom: igen / nem

....
aláírás

Dr. Gerzson Miklós
Elfogadásra javaslom: igen / nem

....
aláírás

A jelölt a doktori szigorlaton .... %-ot ért el.

Az értekezést bírálóként elfogadásra javaslom:

Bíráló neve: .... : igen / nem

....
aláírás

Bíráló neve: .... : igen / nem

....
aláírás

A jelölt az értekezés nyilvános vitáján .... %-ot ért el.
Veszprém, ...

....
a Bíráló Bizottság elnöke

A doktori (PhD) oklevél minősítése: ....

....
az EDHT elnöke

Contents

Acknowledgements

First of all, I wish to express my deepest gratitude to my supervisors, Prof. Katalin Hangos and Dr. Miklós Gerzson, who convincingly guided and encouraged me to be professional and do the right thing even when the road got tough. Without their persistent help, the goal of this project would not have been realized.

I am also grateful to my colleagues Dr. Attila Magyar and Roland Bálint to their continuous support and companionship.

Furthermore I wish to thank to the members of the Department of Electrical Engineering and Information Systems for their encouragement through the years.

Finally I am grateful to my sisters Dia and Melinda, and my significant other András who have provided emotional support during my work. I am also grateful to my other family members and friends who have supported me along the way.

Abstract

Electrical energy is one of the most essential resources of the modern world. Electrical energy systems usually have distributed structures with redundant and reconfigurable elements, which grant the safe and reliable operation of the system. The diagnosis of such a system is important because the failure of a component has effect on the fault tolerance of the whole system. Effective diagnosis methods may reduce the number of equipments and save maintenance costs of a complex system. In this work different kinds of model based diagnostic solutions are presented focused on energetic application areas, where the electrical energy system is modelled and analyzed at different levels of hierarchy.

First a parameter estimation method of batteries is presented, that can be a basis of a diagnostic method. A simple parametric temperature dependent battery model is used for this purpose. A two-step method is used that includes a parameter estimation step of the key parameters at different temperatures followed by a static optimization step that determines the temperature coefficients of the corresponding parameters. The proposed method can be used as a computationally effective way of determining the key battery parameters at a given temperature, that can be used for battery health diagnosis.

Next, a diagnostic method is proposed for detecting and isolating non-technical losses (illegal loads) in low voltage electrical grids. The proposed method uses a simple static linear model of the network and it is based on analyzing the differences between the measured and model-predicted voltages. As a preliminary step of the diagnosis, the decomposition of the network is proposed to make the computation efficient. The uncertainty in the model parameters and measurements are also taken into account to make the approach applicable in real-world cases. The proposed method is able to localize multiple illegal loads in the network.

At last a new on-line fault identification method of technological processes is proposed that uses a qualitative dynamic model of the system and its colored Petri net model. The diagnosis is based on searching the deviations between the traces of the normal and the actual operation on the occurrence graph of the model. In case of composite systems the occurrence graph can be extremely large, therefore a structural decomposition method is applied which can manage increased computational effort and searching-time.

The presented methods can be applied in different areas of electrical system diagnosis. The battery parameter estimation method can be used for estimating battery age and remaining life, the available capacity and state of charge at different temperatures. The decomposition based methods can be used for diagnosing both static and dynamic systems. Using decomposition, multiple fault diagnosis is possible. In addition it may has a positive effect on the computational effort.

Kivonat

A villamos energia napjaink egyik meghatározó energiaforrása. A fogyasztók biztonságos és folyamatos ellátása érdekében, a villamos energetikai rendszerek általában elosztott struktúrájú, számos redundáns illetve rekonfigurálható elemet tartalmazó rendszerek. A diagnosztika egy ilyen rendszerben nagy szerepet kap, mivel az egyes komponensek meghibásodása a teljes rendszer hibatűrő képességét is befolyásolja. Hatékony diagnosztikai módszerek segítségével a berendezések száma és a karbantartási költségek optimalizálhatók. Jelen értekezésben különböző modell alapú diagnosztikai módszerek kerülnek bemutatásra, amelyek villamos energetikai rendszerek különböző szintű vizsgálatához kapcsolódnak.

Először egy akkumulátorok paraméterbecslésére szolgáló módszer kerül bemutatásra, ahol az akkumulátor működését egy egyszerű parametrikus, hőmérsékletfüggő modell írja le. A javasolt kétlépéses paraméterbecslési módszer segítségével az akkumulátor paraméterei és paraméterek hőmérsékletfüggő karakterisztikái hatékonya megbecsülhetők. A módszer felhasználható akkumulátor-diagnosztikai alkalmazásokban, ahol az akkumulátor elhasználódottságára, élettartamára kell következtetni.

Ezután egy, a villamos hálózatokban jelen lévő nem mért vételezés diagnosztikájára szolgáló módszer kerül bemutatásra. A módszer segítségével detektálni és azonosítani lehet a hálózatban jelen lévő illegális fogyasztókat. A javasolt diagnosztikai módszer pedig a mért és a hálózat statikus lineáris modellje által jósolt feszültségek és áramok összehasonlításán alapul. A kifejlesztett módszer egyszerre több illegális fogyasztó azonosítására is képes, valamint figyelembe veszi a hálózat paramétereinek bizonytalanságát, és a mérési hibákat is. A módszer hatékonyságának növelése érdekében a hálózat előzetes dekompozíciója javasolt.

Végül technológiai rendszerek kvalitatív modell alapú diagnosztikája kerül ismertetésre. A diagnosztikai módszer a technológiai rendszer normál és hibás működéseit leíró eseménysorai közötti eltérések vizsgálatán alapszik, felhasználva a rendszer színezett Petri háló modelljének elérhetőségi gráfját. Összetett rendszerek diagnosztikája esetében a rendszer előzetes strukturális dekompozíciója javasolt, a különböző egységekben előforduló hibák könnyebb azonosítása érdekében.

A bemutatott módszerek különböző területeken alkalmazhatóak. Az akkumulátor paramétereinek becslése felhasználható az aktuális és a hátralévő élettartam, az elérhető kapacitás és töltöttségi szint becslésére különböző hőmérsékleteken. A dekompozíció alapú módszerek statikus és dinamikus modellek esetében is alkalmazhatóak. Ezen kívül a dekompozíció lehetővé teszi, hogy többszörös hibák is könnyen azonosíthatóak legyenek, valamint a számítási igényt is kedvezően befolyásolja.

Аннотация

Электрическая энергия — один из самых важных ресурсов современного мира. Системы электрической энергии обычно имеют распределенные структуры с многими избыточными элементами, которые обеспечивают безопасную и надёжную работу системы. Диагностика таких систем очень важная, потому что повреждение одного элемента влияет на отказоустойчивость всей системы. Эффективные методы диагностики могут уменьшить количество оборудований и расходы на техническое обслуживание сложных систем. В этой диссертации представляются разные виды диагностики на основе модели энергетических систем.

Сначала представляется метод оценки параметров аккумулятора, который может быть основа метода диагностики. Для этой цели использована простая модель аккумулятора, зависимая от температуры. Разработанный метод состоит из двух этапов. На первом этапе оцениваются ключевые параметры аккумулятора при разных температурах. На втором этапе оцениваются температурные коэффициенты параметров с помощью статической оптимизации. Предложенный метод можно использовать как вычислительно эффективный способ, чтобы оценить параметры аккумулятора.

После этого, представляется диагностический метод для детектирования и изоляции нетехнических потерь в электрических сетях. Для этой цели использована статическая линеарная модель электрической сети. Предложенный метод основывается на анализе измеренных и предсказанных напряжений. Для того чтобы сделать метод вычислительно эффективным, перед диагностикой предлагается декомпозиция сети. Разработанный диагностический метод может локализовать несколько нелегальных потребителей, учитывая погрешности измерения.

Наконец, предлагается новый онлайн метод диагностики технологических процессов который использует качественную динамическую модель системы в форме цветной сети Петри. В этом случае диагностика основывается на поиске разниц между следами нормальной и реальной операции на графе достижимости. Графы достижимости сложных систем могут быть огромными, поэтому предлагается метод структурной декомпозиции, который может управлять повышенными вычислительными усилиями и временем на поиск.

Представленные методы могут применяться в различных областях диагностики электрических систем. Метод оценки параметров аккумулятора может использоваться для оценки срока службы, емкости или уровня заряда аккумулятора при разных температурах. Методы, основанные на декомпозиции, могут использоваться для диагностики и статических и динамических систем. Декомпозиция позволяет диагностировать множественные неисправности, и, кроме того, положительно влияет на вычислительное усилие.

Chapter 1

Introduction

1.1 Background and motivation

Electrical energy plays a fundamental role in our everyday life. Industrial processes, communication, household tasks, entertainment or transportation in modern days would be inconceivable without electrical energy. The transmission of the electrical energy from the place of generation to the consumption requires complex and extensive transmission and distribution networks, which may even go beyond country borders. Therefore, research done in the area of these transmission and distribution networks and their elements (electrical machines, consumption equipments, batteries, power sources etc.), called as electrical energy systems, are of great importance.

Electrical energy systems are usually distributed systems with high degree of redundancy and reconfigurability. To grant the continuous operability of the system and the availability of the energy, electrical energy systems are usually equipped with backup systems therefore a single fault should not significantly affect the operation of the system. The diagnosis of the system components is important because it affects the fault tolerance of the whole system. An effective diagnostic method may reduce the number of necessary redundant components and additional costs of (e.g. maintenance). Therefore the diagnosis of electrical energy systems is crucial for the maintainers, operators and customers of the system.

The diagnosis of the electrical energy systems can be performed at different levels with different capabilities. Sometimes it is only enough to detect the presence of the fault, but usually the type and the location of the occurred fault should be identified. The situation may become even more complicated if there are multiple faults in the system. Moreover an other advantage of the model based diagnosis is that the model used for the diagnosis can be utilized in other areas of the investigation of the system (e.g. verification).

Electrical energy systems can be modelled and analyzed at different levels of hierarchy. At the top level network model a simple static linear model may be enough to describe the main operation of the network [1]. Going down to lower levels, such as distribution networks, local low-voltage networks, consumers and devices, the models of the components become more complex and detailed and offer higher resolution in space and time [2]. For example a local network can be modelled as a discrete event system [3]; or a low level component, like a battery can be modelled by a nonlinear state space model [4]. When modelling a certain part of the network one selects the appropriate model type according e.g. to the modelling goal, the targeted application area and the available computational resource.

In this work various model based fault diagnostic methods applied to electrical energy systems are presented with different approaches of the diagnostic task. The presented methods vary in the applied method, the used system model and the application area.

1.2 Model based diagnosis in dynamic systems

Each system has an expected way of operation that satisfies the purpose for which it was created. However it can sometimes occur that the system operates incorrectly because of different reasons. The malfunction of the system can be caused by a *fault*. The sign of the fault is the deviation between the actual (measured, observed or computed) and the nominal variables/parameters, that is called *error*. If the error remains unnoticed and the fault is not handled properly, then the situation may lead to the *failure* of the system. A failure occurs when the system can no longer satisfy its original function, or its performance is degraded [5]. Faults may change the structure, variables or the parameters of the system model [6]. Model based diagnosis is one of the most popular group of diagnostic methods [7].

The general diagnostic task can be described in the following way [8]. Given a set of faults $\mathcal{F}$ (where $f_0 \in \mathcal{F}$ refers to the faultless mode), a dynamic system model including the possible fault models and the measured input-output data: $U = (u(0), u(1), \ldots, u(k))$, $Y = (y(0), y(1), \ldots, y(k))$. The diagnostic task is to find the fault f for a given input-output pair (U, Y).

Different levels of the diagnosis can be distinguished based on the depth of resolution. The task of *fault detection* is to decide whether any fault has occurred in the system or not. *Fault isolation* aims to determine the location of the fault within the system. *Fault identification* is used to determine exactly which fault in $\mathcal{F}$ has occurred and estimate its magnitude.

In the following a short review of model based diagnostic methods that are used in this thesis is given.

1.2.1 Diagnosis based on parameter estimation

The basic principle of the method is that the occurrence of a fault makes changes in the variables or parameters of the system. Given the set of faults, the parametrized system model, the measured input-output data, and a loss function depending on the parameters, the task is to find those parameters corresponding to a given fault that minimizes the loss function such that the model output is as close as possible to the measured output [9].

Several methods of parameter estimation can be used to solve this task. One of the most popular methods is the least squares estimation and its variations because of their simplicity.

The parameter estimation based diagnosis is usually applied at component level because of the complexity of the model and the number of parameters. In [10] a parameter estimation based fault detection method was developed for a DC motor where both electrical and mechanical parameters were estimated. Recursive least squares are often used for real-time diagnosis exploiting the reduced computational effort [11]. Nonlinear weighted least squares was used to predict the degradation of a gas turbine [12]. Parameter estimation can also be a part of a hybrid diagnoser as it was applied in [13], [14].

The parameter estimation methods can be applied with high reliability only if the input is properly excited. Experiment design aims to create optimal conditions for parameter estimation. The papers [15] and [16] propose an experiment design solution that is optimal from the parameter identification point of view, where solution space is a sinusoidal signal family applied as charging/discharging current. On the other hand, experiment design [17] can also be used in order to maximize the information content of the battery charging-discharging related measurement dataset in order to estimate battery parameters more precisely.

Parameter estimation of nonlinear models can be computationally complex since the loss function should be minimized numerically. Paper [18] overcomes this problem by a parallel Java algorithm implemented on GPU (CUDA) architecture. The authors of [19] developed and compared three different solutions for the internal resistance estimation of lithium-ion batteries (direct resistance estimation, Extended Kalman Filter (EKF), recursive Least Squares) and concluded that EKF approach performed the best in terms of computational efficiency.

The advantage of the parameter estimation based diagnosis is that all fault diagnosis tasks (detection, isolation, identification) can be realized with this method. Besides that, with on-line parameter estimation methods faults can be recognized at an early stage and the detection of multiple faults is also possible.

1.2.2 Diagnosis based on analytical redundancy

Analytical redundancy based diagnosis methods use the idea that the actual measured variables should be consistent with the model calculated variables [20]. Analytical redundancy relations (ARR) are equations representing the deviations between the measured and model variables that are zero in case of normal operation and nonzero in case of at least one fault.

Analytical redundancy based diagnosers usually have two components. The *residual generator* computes the the difference between measured and model variables (residuals). Usually three different approaches are used for residual generation: the parity space, the observer and the Kalman filter methods. The *decision system* analyzes the residuals and complete the fault detection and isolation tasks [21].

The fault detection task is relatively simple with analytical redundancy. However the residuals being zero in/during normal operation is typically not true in real world applications because of the presence of measurement noise. Therefore the residuals are usually compared to a given threshold to detect faults in the system[22]. When the measurement noise is significant then information about the noise should be included in the model. Using a properly designed filter the effect of noise can be attenuated.

For fault isolation more than one residual is needed. The two main isolation techniques that can be used are the method of structured residuals and the fixed direction residuals [23]. In the first case the residuals can be divided into subsets that are nonzero only if a specific fault has occurred. The pattern of zero and nonzero elements of the residuals, which is called signature, characterizes the fault. In the second case the direction of the residual vector can be associated with the given fault.

Analytical redundancy is often used to diagnose sensor or actuator faults. Application areas range from simple benchmark systems like the three tank system in [24] to complex engineering systems and safety critical systems such as nuclear power plants [25], jet engines [26], satellites [27] or aerospace [28].

The advantage of analytical redundancy based diagnosis in contrast to hardware redundancy is that no duplication of sensors or physical components is needed to realize a diagnostic method. This reduces the cost and weight of the equipment. An other feature is that a variety of models can be served as the basis of the diagnosis (e.g.ordinary differential equations, data-driven models, expert systems) [29].

1.2.3 Diagnosis of discrete event systems

Discrete event systems are special kinds of dynamic systems with discrete time and discrete valued variables. Events are the changes between the discrete values of the variables. Typical models of discrete event systems are automata models and their extensions, Petri nets or state machines.

Fault diagnosis of discrete event systems is usually based on the assumption that faults are unobservable events, therefore only the effects of faults can be noticed [30]. Since events related to faults are unobservable by assumption, the detection and isolation problem must be solved based on the available information of the observed non-faulty events. The diagnosability of discrete event systems has been originally investigated in [31] using the methods of automata theory.

If the discrete event system is modelled by an automaton then the most common solution of the diagnostic problem is the creation of the diagnoser automaton by eliminating the unobservable events. Although the diagnoser can be constructed algorithmically, the issues of state explosion and high computational complexity are present. A solution to reduce the computational effort was presented in [32].

Another way of diagnosis is performed by checking the consistency of the observed input-output sequences and the state transition relations of the automaton model [33]. The method can be applied to quantised systems represented by discrete event models, too [34].

Discrete event systems can be represented by different kinds of Petri nets, too. A simple fault detection method based on the measurement of token quantity in conservative Petri nets is given in [35]. In case of more complex models more sophisticated diagnostic methods are needed. The construction of a diagnoser Petri net, which is the copy of the original model without the faulty transitions, is proposed in [36], [37]. Comparing the original model output and the diagnoser output the difference between them indicates that a fault has occurred. Besides that, the reachability or the coverability graph of the Petri net is often used for diagnostic purposes, because it contains all possible system states [38]. Other methods use the mathematical model of the Petri net and the diagnostic problem is traced back to the solution of a set of linear equations [39], [40]. Qualitative discrete event systems, which is in the focus of Chapter 4, can be also modelled by ordinary or colored Petri nets [41], [42]. The trace of the qualitative variables can be used for detecting and identifying faults in the system, using a specially constructed colored Petri net diagnoser [43].

The problems of the Petri net based diagnostic methods are similar to the ones mentioned at the automata based methods.

1.3 Thesis structure

This thesis is divided into the following main chapters.

In Chapter 2 the parameter estimation method developed for the diagnosis of batteries is presented. First the basic notions related to the battery operation are introduced then the used temperature dependent battery model is introduced. After that the experiment design process that is a necessary prerequisite of the parameter estimation is described. The developed two-step parameter estimation method is explained hereinafter. Finally, simulation results are presented and the achieved results are summarized.

In Chapter 3 the model based diagnosis of electrical networks is described which is based on the comparison of nominal and measured current/voltage values. First the model of the network and the basic concepts are introduced. Then the decomposition of different network structures is presented which is the key element of the diagnostic method. After that the proposed diagnostic algorithms for non-technical loss diagnosis utilizing the network structures are presented. The decomposition and diagnostic methods are illustrated on simple examples. The results are summarized at the end of the chapter.

In Chapter 4 the colored Petri net based diagnostic method for process systems is presented. At first the basic notions of ordinary and colored Petri nets are introduced. The general colored Petri net model and its operation is explained in the next section. After that the basic unit-wise diagnostic algorithm is presented which uses the occurrence graph of the colored Petri net model of the technological unit. The diagnosis of composite systems based on structural decomposition is introduced in the next section. Finally, the results of this work are summarized.

In Chapter 5 the main results of this research are summarized in three thesis points. The relevant publications can also be found here.

Two detailed case studies illustrating the diagnostic methods presented in Chapter 3 and Chapter 4 can be found in the Appendix.

Chapter 2

Diagnosis of batteries using parameter estimation

Lithium-ion batteries are popular energy sources of our everyday life because of their high energy density, low self-discharge and light weight. Portable electronic devices (mobile phones, laptops), home electronics, electronic tools and electric vehicles (EVs) all run on some type of lithium-ion battery. In applications like electrical vehicles, batteries are connected in parallel and series in order to meet the power needs.

The optimal performance and safe operation of the set of battery cells are managed by the battery management system (BMS). Another essential role of the BMS is the state of charge (SOC) and state of health (SOH) estimation. The former quantity informs the user on the remaining charge of the battery bank (i.e. the remaining mileage that can be travelled with the electrical vehicle), while the latter shows the ratio between the capacity of a new battery in relation to the actual capacity of the battery. Just like any other battery, the performance of the lithium-ion battery is not constant but slowly degrades during the operation and strongly depends on the ambient temperature. The battery health conditions cannot be measured directly therefore it should be estimated based on measurable quantities.

The temperature does not only affect the aging process of a battery but its short term performance, too. Therefore the thermal effects taking place in the battery should also be taken into account when creating a battery model.

Thermal modelling and the analysis of lithium-ion batteries under different temperatures has been addressed by several authors. The thermal modelling of batteries as well as the modelling without temperature dependency can be classified based on the scientific background (e.g. equivalent circuit models, electrochemical models). The review [44] gives a thorough analysis not only of the different electrochemical models but also of the parameter identification methods.

In such applications where the computational complexity (i.e. time) is crucial e.g. in BMSs, *equivalent circuit models* are widely used [45]. In our previous work [46] we proposed a parameter estimation for lithium ion batteries based on their first order equivalent circuit model. The results showed that there is a linear dependency between some model parameters because of the insufficient excitation.

The aim of the work presented here is to diagnose the failure of the battery related to the aging. Because the aging process affects the battery parameters (e.g. decreased capacity), the condition of the battery can be estimated by comparing the actual battery parameters with the parameters of a new battery. To do this the actual battery parameters should be determined. In the following sections I propose a parameter estimation method which takes into account the temperature dependency of the parameters, too. The temperature dependent battery operation is critical in automotive applications (e.g. the available charge in a battery at 0 $°C$ is less than at 25 $°C$).

In this chapter I generalize my previous work to the case when the temperature dependence of the parameters are also taken into account. The way of estimation of the temperature dependent parameters is important from the viewpoint of applicability a simple parameter estimation is needed for primarily the actual capacity for implementation in future BMSs. In this work a different equivalent circuit model is used instead of the one in [46], with physical meanings of the parameters. The novelty of this work is the two-step parameter estimation method of the temperature dependent parameters. Before the estimation of the parameters, proper experiment design was carried out to ensure sufficient excitation and maximize the available information about the parameters. The proposed method can be used as a basis of a battery health diagnosis system.

2.1 Battery basic notions

In this section the basic notions related to the operation of Li-ion batteries are summarized according to [47].

The *battery* is an electrochemical energy source that is composed of rechargeable cells. The battery delivers voltage that depends on the cell chemistry. The components of an electrochemical cell are the positive electrode , the negative electrode and the electrolyte. In case of Li-ion battery the positive electrode (cathode) is made from metal oxide, the negative electrode (anode) is made from carbon (graphite) and the electrolyte is a liquid solvent with lithium salt. The schematic structure of a Li-ion battery can be seen in Figure 2.1.

Batteries have two operation modes based on the direction of the current: *discharge* and *charge.* During discharge the positive Li-ions move from the negative electrode to the positive electrode through an external load. During charge an external electrical power source is connected to the electrodes that causes over-voltage. This forces the positive ions to move from the positive electrode to the negative electrode.

The *capacity* of a battery is the amount of electric charge that a fully charged battery can deliver during discharge. The capacity is defined as the product of the constant discharge current and the time, and it is measured in ampere-hours (Ah). For example a battery with 20 Ah capacity provides 20 A current for 1 hour.

The *nominal capacity* is the capacity of the battery when it is discharged with the nominal current provided by the manufacturer.

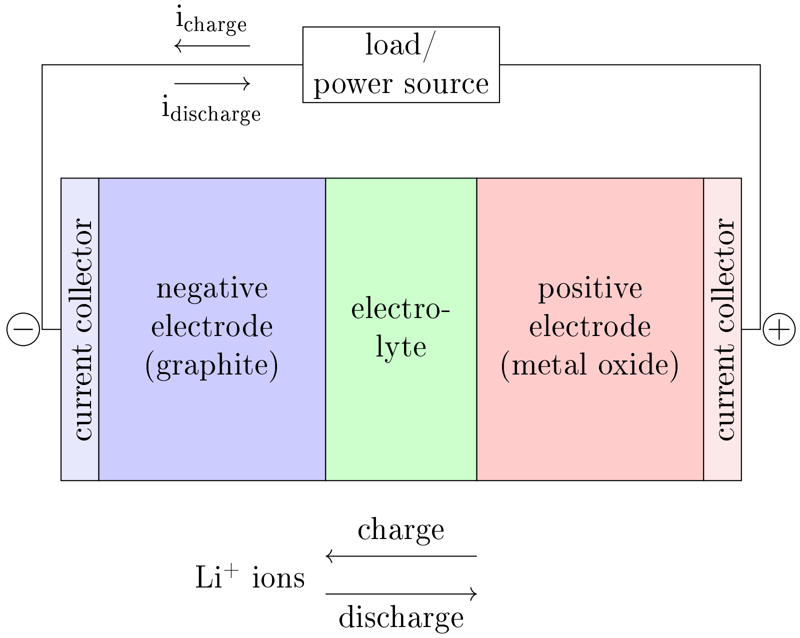


Figure 2.1: The schematic structure of a Li-ion battery

The charge/discharge current is often expressed with the *C rate* instead of the actual magnitude. The C rate is a relative quantity that is defined as the charge/discharge current related to the nominal capacity of the battery. For example a 20 Ah battery can be discharged with 1 C (20 A) in 1 hour, or with 0.1 C (2 A) in 10 hours.

The *State of Charge (SOC)* is the actual available charge in the battery. It is usually expressed as a percentage of the nominal capacity.

$$SOC(t) = \frac{Q(t)}{Q_{nom}} \cdot 100$$

The SOC is 100 % for a fully charged battery and 0 % for a fully discharged battery. The SOC is a crucial information of the battery therefore accurate SOC estimation is a major object in battery related researches [48]–[52].

The *open circuit voltage (OCV)* of the battery is the voltage between the battery terminals when there is no current flow, i.e. the battery is neither charged or discharged. The open circuit voltage is affected by several factors, for example state of charge, temperature and polarization.

Polarization is a side effect in batteries that occurs when the electrode potential is displaced from the equilibrium potential due to a passage of current through the cell. This effect is slowly developing over time and affects the open circuit voltage.

The *terminal voltage* of the battery is the potential difference between the battery terminals (denoted by + and - signs in Figure 2.1). The battery terminal voltage is smaller than the OCV during discharge and greater than the OCV during charge. This phenomenon is caused by the *internal resistance*.

All materials have some electrical resistance including the components of the battery cells. The internal resistance of a battery comes from the resistivity of the electrodes and the electrolyte.

The battery capacity degradation during usage is usually characterized by the *State of Health (SOH)*. The SOH of a battery is measured as the battery capacity related to the initial nominal capacity in percentages:

$$SOH(t) = \frac{Q_{nom}(t)}{Q_{nom}(t_0)} \cdot 100$$

The battery reaches its end of life and considered degraded when the SOH drops below 80 %. Different methods of SOH estimation are presented in the literature [48], [49], [53], [54].

2.2 Battery model

The parametric lithium-ion battery model that is the basis of the methods to be proposed later is presented here. This is a modified version of our model used in [46]. The list of notations used in the battery model is given in Table 2.2.

2.2.1 Modelling assumptions

The following assumptions were made for the battery model [55] with temperature dependency:

- The capacity of the battery does not change respective to amplitude of the current (no Peukert effect).
- The self-discharge of the battery is not represented.
- The memory effect is less important from the viewpoint of ageing than discharge/recharge strategy/policy.
- The voltage and the current can be influenced.
- The capacity depends on the ambient temperature.
- The electrode potential, the polarization coefficient, the polarization resistance and the internal resistance depend on the internal (cell) temperature of the battery.

2.2.2 Temperature dependent battery model

From the potential modelling methodologies the equivalent electrical circuit type was selected to create the basic battery model. The selected model is originally developed in [55], in our previous work we described that model without temperature effect [56].

The structure of the model can be seen in Figure 2.2.

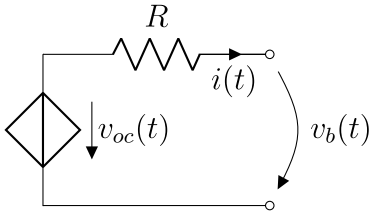


Figure 2.2: Equivalent electrical circuit model of the battery. Voltage $v_{oc}(t)$ of the controlled voltage source is different in the case of charge and discharge.

The input of the model is the battery current (i) and the output is the battery terminal voltage (v_b). The open circuit voltage (v_{oc}) is represented by a controlled voltage source, and it is different during charge and discharge. The model was extended with temperature effects as it can be found in the Matlab Simulink Battery block (*Simulink/Simscape/Electrical/Specialized Power Systems/Electric Drives/Extra Sources*) [57]. The difference with respect to the basic model [56] is that some of the parameters depend on the ambient or cell temperature. As a result, the temperature dependent state space model of the battery is obtained in the form of Eqs.(2.1-2.6) following [58] with the notations collected in Table 2.1.

State equations:

$$\frac{d}{dt}q(t) = \frac{1}{3600}i(t) \tag{2.1}$$

$$\frac{d}{dt}i^*(t) = -\frac{1}{\tau}i^*(t) + \frac{1}{\tau}i(t) \tag{2.2}$$

The state variables have the following meaning:

- q is the actual extracted capacity of the battery. The initial values are $q(t_0) = 0$, if the battery is fully charged and $q(t_0) = Q$, if the battery is fully discharged.
- i^* is the polarization current. It can be computed by applying a low-pass filter to the battery current i, where τ is the time constant of the filter (see Eq. (2.2)).

Output equations:

- Charge model

$$\begin{aligned} v_{oc}^{ch}(t,T,T_a) =& E_0(T) - K_1(T)\frac{Q(T_a)}{q(t)+0.1Q(T_a)}i^*(t) - \\ &- K_2(T)\frac{Q(T_a)}{Q(T_a)-q(t)}q(t) + A\exp(-Bq(t)) - Cq(t) \end{aligned} \tag{2.3}$$

$$v_b^{ch}(t,T) = v_{oc}^{ch}(t,T,T_a) - R(T)i(t) \tag{2.4}$$

- Discharge model

$$\begin{aligned} v_{oc}^{dch}(t,T,T_a) =& E_0(T) - K_1(T)\frac{Q(T_a)}{Q(T_a)-q(t)}i^*(t)- \\ & - K_2(T)\frac{Q(T_a)}{Q(T_a)-q(t)}q(t) + A\exp(-Bq(t)) - Cq(t) \end{aligned} \tag{2.5}$$

$$v_b^{dch}(t,T) = v_{oc}^{dch}(t,T,T_a) - R(T)i(t) \tag{2.6}$$

The *output* of the model is the battery terminal voltage v_b^X that is composed of the open circuit voltage (v_{oc}^X) and the voltage drop across the internal resistance ($R(T)i(t)$). $X = \{ch, dch\}$ denotes the charge/discharge mode of the battery.

The charge and discharge model differs in the open circuit voltage equation. The open circuit voltage is composed of five main parts. E_0 is the electrode potential of the battery, The term $K_1(T)\frac{Q(T_a)}{q(t)+0.1Q(T_a)}i^*(t)$ and $K_1(T)\frac{Q(T_a)}{Q(T_a)-q(t)}i^*(t)$ represents the polarization phenomenon in case of charge and discharge respectively, where K_1 is the polarization resistance and Q is the battery capacity. The term $K_2(T)\frac{Q(T_a)}{Q(T_a)-q(t)}q(t)$ describes the nonlinear variation of the OCV with the SOC, where K_2 is a polarization constant. The fourth term $A\exp(-Bq(t))$ represents the rapid increase of the battery voltage when the battery is nearly fully charged. Finally $Cq(t)$ represents linear component of the discharge curve of the battery.

The variables and parameters of the model with their meaning and units can be seen in Table 2.1.

The indirect temperature dependency of the model defined by Eqs.(2.1-2.6) is realized through a static temperature dependence of the model parameters. The *temperature dependency of the parameters* can be described with the following equations [58]:

- The change of polarization coefficient, polarization resistance and internal resistance with the battery temperature T can be derived from the Arrhenius law:

$$K_1(T) = K_1|_{T_{ref}} \exp\left(\alpha_1\left(\frac{1}{T} - \frac{1}{T_{ref}}\right)\right) \tag{2.7}$$

$$K_2(T) = K_2|_{T_{ref}} \exp\left(\alpha_2\left(\frac{1}{T} - \frac{1}{T_{ref}}\right)\right) \tag{2.8}$$

$$R(T) = R|_{T_{ref}} \exp\left(\beta\left(\frac{1}{T} - \frac{1}{T_{ref}}\right)\right) \tag{2.9}$$

- The temperature dependency of the capacity and the constant potential can be written in the following form:

$$Q(T_a) = Q|_{T_{ref}} + \frac{\Delta Q}{\Delta T}(T_a - T_{ref}) \tag{2.10}$$

$$E_0(T) = E_0|_{T_{ref}} + \frac{\partial E}{\partial T}(T - T_{ref}) \tag{2.11}$$

Table 2.1: Variables and parameters of the examined Samsung INR18650-20Q Li-ion battery

Name	Type	Meaning	Unit	Value
i	input variable	battery current	A	-
i^*	state variable	polarization current	A	-
q	state variable	extracted capacity	Ah	-
t	independent variable	time	s	-
v_{oc}	variable	open circuit voltage	V	-
v_b	output variable	battery voltage	V	-
T	external variable	battery cell temperature	K	-
T_a	external variable	ambient temperature	K	-
τ	parameter	time constant of the filter	s	0.003
E_0	parameter	constant potential of the electrodes	V	-
$\partial E/\partial T$	parameter	reversible voltage temperature coefficient	V/K	0.002
R	parameter	internal resistance	Ω	-
β	parameter	Arrhenius rate constant for the internal resistance	K	3839.8
K_1	parameter	polarization resistance	Ω	-
α_1	parameter	Arrhenius rate constant for the polarization coefficient	K	8415.3
K_2	parameter	polarization constant	V/Ah	-
α_2	parameter	Arrhenius rate constant for the polarization resistance	K	8415.3
Q	parameter	battery capacity	Ah	-
$\Delta Q/\Delta T$	parameter	maximum capacity temperature coefficient	Ah/K	0.016
A	parameter	exponential voltage	V	0.1589
B	parameter	exponential capacity	$(\text{Ah})^{-1}$	15.0
C	parameter	nominal discharge curve slope	V/Ah	0.2362

The parameters of the temperature dependent battery model with their meaning and nominal values can be also found in Table 2.1. Our examined battery is a Samsung INR18650-20Q type battery with 2000 mAh nominal capacity and 3.6 V nominal voltage. The nominal parameters of the battery were extracted from the datasheet and the Matlab Simulink model [59]. The nominal values of the temperature dependent parameters at reference temperature can be found in Table 2.2.

Table 2.2: Parameters of the examined Samsung INR18650-20Q Li-ion battery at reference temperature.

Name	Type	Meaning	Unit	Value
T_{ref}	parameter	nominal ambient temperature	K	298.15
$E_0\|_{T_{ref}}$	parameter	constant potential of the electrodes at nominal ambient temperature	V	3.9388
$R\|_{T_{ref}}$	parameter	internal resistance at nominal ambient temperature	Ω	0.005
$K_1\|_{T_{ref}}$	parameter	polarization resistance at nominal ambient temperature	Ω	0.0018
$K_2\|_{T_{ref}}$	parameter	polarization constant at nominal ambient temperature	V/Ah	0.0018
$Q\|_{T_{ref}}$	parameter	battery capacity at nominal ambient temperature	Ah	2.0

Remark on the battery cell temperature
In order to obtain a simple model for parameter estimation, we have omitted the energy balance and considered the battery cell temperature T as an *external variable*. Simulation results showed that the cell temperature changed about $+2\ ^{\circ}C$ with respect to the ambient temperature during a charge or discharge operation.

2.3 Experiment design

The quality of the estimated parameters depends on the quality of the available measurement data. With the help of experiment design the available information about the parameters that can be extracted from the measurements can be influenced. Techniques of experiment design include choosing the proper input variables, the parameters to be estimated, creating input signals to ensure sufficient excitation, fixing experimental conditions etc. In this section two steps of experiment design are executed. In Section 2.3.1 the parameters to be estimated are chosen based on the sensitivity analysis of the model. Then the input signals used in the experiments are introduced in Section 2.3.2. The simulation setup is described in Section 2.3.3.

2.3.1 Parameter sensitivity analysis

As a first step of the parameter estimation, the parameter sensitivity of the charge and discharge model of the battery has been analyzed. It is important to note, that the temperature has an indirect effect on the model output through the parameters which directly depend on the temperature. Instead of applying the classical methods of sensitivity analysis involving sensitivity equations, the method described in our previous work [46] was used for the sensitivity analysis. In this method the parameter values were changed one by one with ±10% with respect to the nominal values, then the difference between the nominal and the perturbed model was evaluated using a quadratic loss function:

$$W_s(\tilde{\theta}) = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{2}\left(v_b(\theta;k) - v_b(\tilde{\theta};k)\right)^2 \tag{2.12}$$

where θ denotes the parameter vector, and $\tilde{\theta}$ is the perturbed parameter vector. At first the step response of the model was simulated to get the time constant of the system (τ_s). The sample time of the PRBS signal (T_s) was chosen to be $T_s = \tau_s/5$. The sensitivity analysis was repeated at 6 different temperatures: $0°C$, $10°C$, $20°C$, $30°C$, $40°C$ and $50°C$. The battery was charged/discharged between $0 - 100\%$ state of charge with PRBS current input (amplitude: charge {-2 A, -0.5 A}, discharge {0.5 A, 2 A}, sample time: 160 s). Both the charge and the discharge models were analyzed. The nominal model was the charge/discharge model at the nominal ambient temperature $T_{ref} = 25°C$.

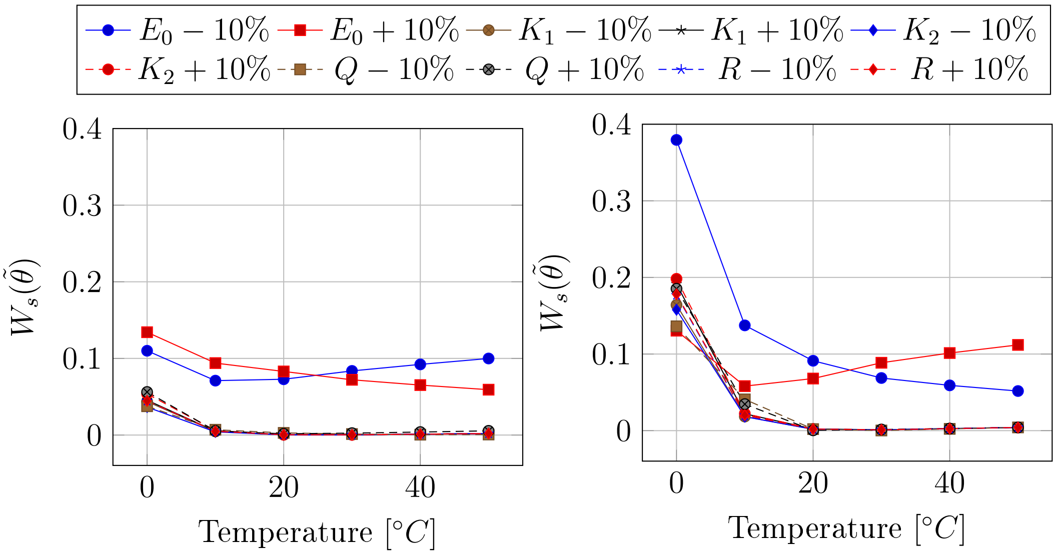
The models were simulated in Matlab using the model equations Eqs. (2.1-2.6). At each temperatures the nominal parameters were perturbed one-by-one and the value of the loss function was computed. The result of the sensitivity analysis of the charge and the discharge model can be seen in Table 2.3 and Table 2.4. The graphical representation of the results is depicted in Figure 2.3.

Table 2.3: Values of the loss function in case of the parameter sensitivity analysis of the charge model.

Parameter	Change	$0°C$	$10°C$	$20°C$	$30°C$	$40°C$	$50°C$
E_0	-10%	0.1100	0.0710	0.0728	0.0837	0.0922	0.0999
	+10%	0.1342	0.0939	0.0830	0.0721	0.0652	0.0592
K_1	-10%	0.0437	0.0047	0.0003	0.0003	0.0011	0.0020
	+10%	0.0455	0.0051	0.0004	0.0003	0.0011	0.0020
K_2	-10%	0.0365	0.0041	0.0003	0.0003	0.0011	0.0020
	+10%	0.0537	0.0059	0.0004	0.0003	0.0011	0.0020
Q	-10%	0.0376	0.0069	0.0028	0.0013	0.0005	0.0007
	+10%	0.0562	0.0054	0.0016	0.0025	0.0039	0.0055
R	-10%	0.0446	0.0049	0.0003	0.0003	0.0011	0.0020
	+10%	0.0446	0.0049	0.0004	0.0003	0.0011	0.0020

Table 2.4: Values of the loss function in case of the parameter sensitivity analysis of the discharge model.

Parameter	Change	$0°C$	$10°C$	$20°C$	$30°C$	$40°C$	$50°C$
E_0	-10%	0.3795	0.1374	0.0912	0.0687	0.0591	0.0517
	+10%	0.1305	0.0581	0.0680	0.0886	0.1013	0.1119
K_1	-10%	0.1641	0.0184	0.0018	0.0011	0.0026	0.0042
	+10%	0.1913	0.0220	0.0022	0.0011	0.0026	0.0042
K_2	-10%	0.1578	0.0182	0.0017	0.0011	0.0026	0.0042
	+10%	0.1982	0.0223	0.0023	0.0010	0.0026	0.0042
Q	-10%	0.1362	0.0408	0.0020	0.0002	0.0023	0.0042
	+10%	0.1852	0.0346	0.0004	0.0015	0.0027	0.0042
R	-10%	0.1769	0.0200	0.0200	0.0011	0.0026	0.0042
	+10%	0.1780	0.0203	0.0020	0.0011	0.0026	0.0042



(a) Sensitivity of the charge model (b) Sensitivity of the discharge model

Figure 2.3: Results of the parameter sensitivity analysis of the charge and the discharge model

It can be seen that the discharge model is a bit more sensitive to the change of the parameters as the magnitude of the error is greater in that case. Both the charge and the discharge models have similar characteristics with respect to the parameter sensitivity:

- The models are highly sensitive to the constant potential E_0.
- The models are less sensitive to K_1, K_2 and Q.
- The rate of sensitivity is similar in case of K_1, K_2 and Q.

- The sensitivity of the models increases as the temperature decreases.
- At ambient temperatures greater than the nominal temperature, the effect of changing the parameters is really small (except for E_0), especially in case of the discharge model.
- The change of the internal resistance R at different temperatures has no effect on the models, as the errors related to the ±10% change are almost the same. In these cases only the temperature affects the models.

Based on these statements *the parameters E_0, K_1, K_2 and Q will be estimated while R is fixed to its nominal value.*

2.3.2 Input signal

The pseudo-random binary sequence (PRBS) is chosen as the input signal for the parameter estimation. It is a widely used signal in the field of parameter estimation [60] because it is easy to generate and provides sufficient excitation. The PRBS has only two values in between the signal changes randomly. The two parameters of the PRBS are the range (the upper and lower level of the signal) and the frequency of the change that should be chosen considering the system dynamics. In our case the clock frequency of the PRBS was chosen to be 5 times the time constant of the system, the latter can be approximately determined from the step response of system (see in Section 2.3.1).

An other important factor of our parameter estimation method is the ambient temperature. The experiments were carried out at different ambient temperatures that were hold constant during an experiment.

The minimum and maximum ambient temperatures of the experiments were chosen according to the recommended operating temperatures of the examined battery. Then this range was evenly divided to get the list of ambient temperatures at which the experiments were carried out. For example if the operating temperature range of the battery is $[0°C, 50°C]$ then the experimental ambient temperatures can be $0°C$, $5°C$, $10°C$, $\ldots, 50°C$.

A method for optimal design of experiments was developed in our previous work [56]. In that paper two different input signals (CCCV (Constant Current Constant Voltage) and PRBS charge/discharge profiles) were investigated in order to maximize the available information about the parameters to be estimated.

2.3.3 Simulation setup

The parameter estimation methods were implemented and tested by simulation experiments in Matlab. To simulate the heat dissipation of the battery during charge/discharge, the battery model in *Simulink/Simscape/ Electrical/Specialized Power Systems/Electric Drives/Extra Sources* (an *extended model*) was used[57]. *This model contains additional energy balance equations* that describe the temperature effects of the battery [61]. This means that the cell temperature and the heating/cooling effects of the battery (including self-heating) during the operation can be simulated. It is important to note that the model used for parameter estimation Eq. (2.1-2.11) is much more simple, as it does not contain the internal energy balance equation. The advantage of the Simulink model is that *the battery cell temperature can be directly extracted from the model*, which can be used as measurement data for the cell temperature.

The simulated battery was a Samsung INR18650Q-20Q battery with 2000 mAh capacity whose nominal parameters can be seen in Table 2.1 and Table 2.2. The operating temperature range of the battery from the datasheet is $[0°C, 50°C]$ for charge and $[-20°C, 75°C]$ for discharge. Based on these values, the ambient temperature was set to be between $0 - 50°C$ for the simulation. The charge and the discharge of the battery was simulated at 11 different ambient temperatures with PRBS input signal between 1-99% state of charge. The simulation setup in case of charge and discharge can be seen below.

Simulation setup for charge:

- PRBS input: $I_{min} = -2A$, $I_{max} = -0.5A$, $T_s = 160s$;
- initial values: $q(t_0) = 0.99Q$, $i^*(t_0) = 0$, $T = T_a$;
- ambient temperatures: $T_a = 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50°C$;
- stopping criterion: $q(t) = 0$.

Simulation setup for discharge:

- PRBS input: $I_{min} = 0.5A$, $I_{max} = 2A$, $T_s = 160s$;
- initial values: $q(t_0) = 0.01Q$, $i^*(t_0) = 0$, $T = T_a$;
- ambient temperatures: $T_a = 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50°C$;
- stopping criterion: $q(t) = 0.99Q$.

All the simulations were performed on a PC (Intel i5 CPU with 4 GB RAM).

2.4 Methods for parameter estimation

The proposed parameter estimation method consists of two steps. At first the battery is charged or discharged at different constant ambient temperatures. At each temperature the parameters E_0, K_1, K_2 and Q of the battery are estimated. In the second step the temperature coefficients of these parameters are estimated using the estimated values of the parameters at different temperatures.

2.4.1 Estimation of the battery parameters

The first step of the proposed method is the estimation of the battery parameters at *different constant ambient temperatures* to see how these parameters change with the temperature. The inputs of the parameter estimation are the battery current and voltage at different temperatures during a full charge or discharge process. The result of the estimation is a set of battery parameters at different temperatures.

It can be seen from Eqs. (2.1-2.6) that the battery model has a nonlinear output equation and four parameters to be estimated. The internal resistance R was fixed to its nominal value because the sensitivity analysis (Section 2.3.1) showed that the model is not sensitive to the parameter R. To estimate the parameters of the model a suitable nonlinear parameter estimation method should be chosen. In this work the nonlinear least-squares method was used for parameter estimation.

Nonlinear parameter estimation problems are usually solved as nonlinear optimization problems where a suitable cost function should be minimized. In our case the cost function is the sum of squared deviation between the model and the measurement data at every time instance (see Eq. (2.13) below).

$$W(\theta) = \sum_{k=1}^{N} \left(\hat{v}_b^X(k) - v_b^X(\theta; k) \right)^2 \tag{2.13}$$

$$X \in \{ch; dch\}$$

where $\hat{v}_b^X(k) = \hat{v}_b^X(kT_s)$ is the measured value of the battery voltage at the k-th sample, $v_b^X(\theta; k)$ is the output of the model (Eq. (2.4) or Eq. (2.6)) with the parameter vector $\theta = [E_0, K_1, K_2, Q]$, and N is the total number of samples.

As all of the parameters to be estimated have physical meaning, the range and scale of the parameter values are usually known in advance. Therefore upper and lower bounds for the parameters can be defined that is useful to limit the searching space of the optimization. As a result, a constrained nonlinear optimization problem should be solved. From the potential algorithms the Trust Region Reflective algorithm [62] was chosen in our work.

2.4.2 Estimation of the temperature dependency of the parameters

The second step of the parameter estimation method is the estimation of the *reference values* and the *temperature dependency coefficients* of the parameters. The inputs of this parameter estimation problem are the estimated parameters at different temperatures from the previous step (Section 2.4.1). It can be seen from the temperature dependent battery model, that the parameters can be divided into two groups based on the type of their temperature dependency:

- Parameters with linear temperature dependency: E_0, Q;

- Parameters with nonlinear (exponential) temperature dependency: K_1, K_2.

Moreover it can be seen from Eqs. (2.7-2.11) that some of the parameters (Q) depend on the ambient temperature and others (E_0, K_1, K_2) depend on the battery cell temperature. The problem is that the cell temperature is not always measurable. In that case when the temperature is measured, the temperature sensor is usually placed on the surface of the battery. In addition the charge/discharge current affects the heat generation of the battery, too. Charging/discharging the battery at low C rates (0.5 C - 1 C rates) we found, that the battery temperature does not increase significantly (2°C). The rapid increase in the battery cell temperature can be experienced when charging/disharging the battery with high C rates (5-25 C) [63]. To overcome these issues the following additional assumptions were made:

- The charge/discharge current of the battery is maximum 1 C (2 A), therefore the cell temperature does not change a lot during charge/discharge (maximum ±2°C).
- The cell temperature is substituted by the average surface temperature during charge/discharge.
- Initially the cell temperature and the ambient temperature are equal ($T(0) = T_a$).
- The surface temperature of the battery is measured.

With the above assumptions the temperature coefficients of the parameters can be estimated. The *coefficients to be estimated* are:

- $E_0|_{T_{ref}}$ and $\partial E/\partial T$ for the temperature dependency of E_0;
- $Q|_{T_{ref}}$ and $\Delta Q/\Delta T$ for the temperature dependency of Q;
- $K_1|_{T_{ref}}$ and α_1 for the temperature dependency of K_1;
- $K_2|_{T_{ref}}$ and α_2 for the temperature dependency of K_2.

The coefficients of $E_0(T)$ and $Q(T_a)$ can be estimated with the simple linear least squares method because equations Eq. (2.11) and (2.10) are linear.

The coefficients of $K_1(T)$ and $K_2(T)$ can also be estimated by the least squares method by transforming the equations and their dependent variables.

2.5 Simulation results

In this section the results of the simulation based experiments are introduced and analyzed. In Section 2.5.1 the results of the estimation of the battery parameters at 11 different temperatures are presented. Then in Section 2.5.2 the results of the estimation of the temperature dependency of the battery parameters are described.

2.5.1 Estimated battery parameters

The battery parameters at different temperatures were estimated using the *lsqnonlin* function from Matlab Optimization Toolbox [64]. This function needs at least two input arguments: a function to minimize and the vector of initial parameter values. Additional input arguments such as lower and upper bounds of the parameters and other options can be also defined. In our case the following bounds were defined for the parameters:

$$\begin{array}{rcccl} 0 & \leq & E_0 & \leq & 5, \\ 0 & \leq & Q & \leq & 3, \\ 0 & \leq & K_1 & \leq & 0.1, \\ 0 & \leq & K_2 & \leq & 0.1. \end{array}$$

The function to minimize is the cost function in Eq. (2.13) and the parameters to be estimated are $\theta = [E_0, Q, K_1, K_2]^T$. The initial values of the parameters were set to the nominal parameter values (see in Table 2.1).

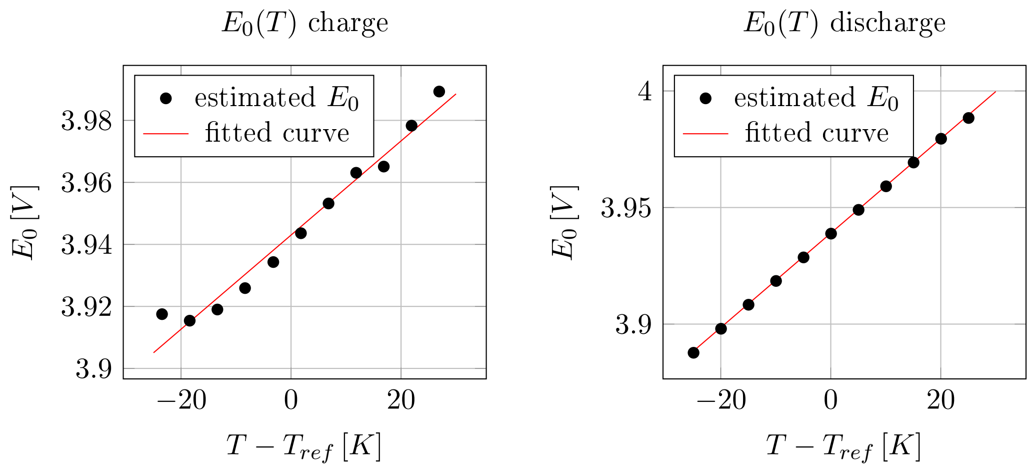
The results of the parameter estimation can be seen in Table 2.5 and Table 2.6. The results are also depicted in Figures 2.4 -2.7 with black dots. It can be noticed in Figure 2.5a that above 35 $°C$ ($T - T_{ref} = 10$) the battery reached its maximum capacity during charge.

Table 2.5: Estimated battery parameters at different temperatures during charge.

$T_a\,[°C]$	0	5	10	15	20	25	30	35	40	45	50
$E_0\,[V]$	3.9175	3.9154	3.9190	3.9259	3.9343	3.9436	3.9532	3.9631	3.9651	3.9783	3.9893
$Q\,[Ah]$	1.6001	1.6800	1.7599	1.8399	1.9201	2.0004	2.0811	2.1623	2.1576	2.1579	2.1582
$K_1\,[\Omega]$	0.0169	0.0099	0.0059	0.0036	0.0023	0.0015	0.0010	0.0007	0.0012	0.0008	0.0007
$K_2\,[V/Ah]$	0.0246	0.0140	0.0082	0.0049	0.0030	0.0019	0.0012	0.0008	0.0000	0.0000	0.0000

Table 2.6: Estimated battery parameters at different temperatures during discharge.

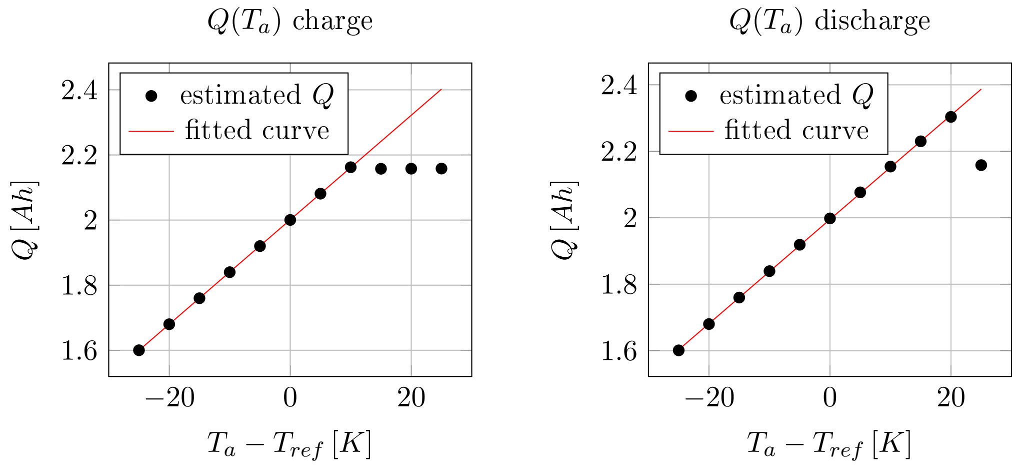
$T_a\,[°C]$	0	5	10	15	20	25	30	35	40	45	50
$E_0\,[V]$	3.8877	3.8980	3.9083	3.9185	3.9286	3.9388	3.9490	3.9591	3.9693	3.9795	3.9884
$Q\,[Ah]$	1.6010	1.6801	1.7599	1.8393	1.9188	1.9980	2.0764	2.1540	2.2300	2.3035	2.1583
$K_1\,[\Omega]$	0.0239	0.0138	0.0081	0.0048	0.0029	0.0018	0.0011	0.0007	0.0004	0.0003	0.0000
$K_2\,[V/Ah]$	0.0243	0.0139	0.0081	0.0048	0.0029	0.0018	0.0011	0.0007	0.0005	0.0003	0.0000



(a) The fitted thermal characteristics of parameter $E_0(T)$ from the charge data.

(b) The fitted thermal characteristics of parameter E_0 from the discharge data.

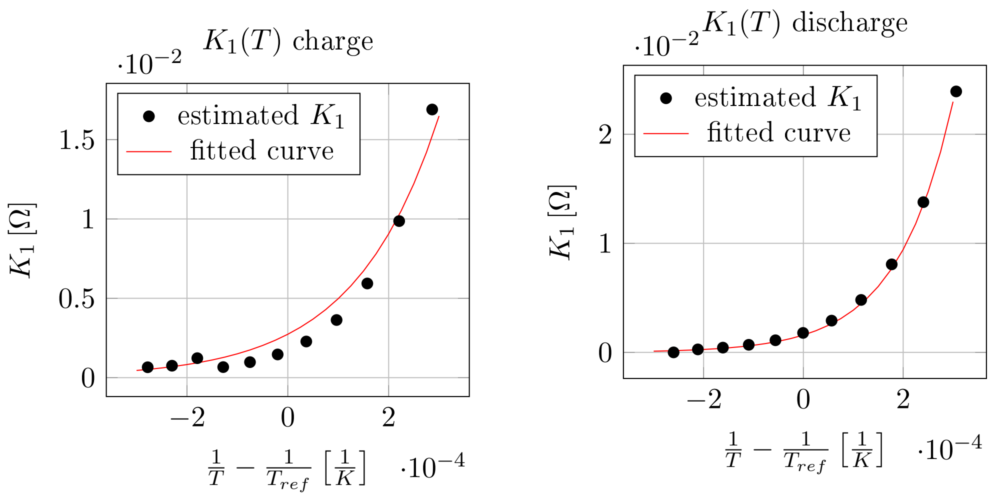
Figure 2.4: Estimation of the temperature dependency of E_0



(a) The fitted thermal characteristics of parameter Q from the charge data.

(b) The fitted thermal characteristics of parameter Q from the discharge data.

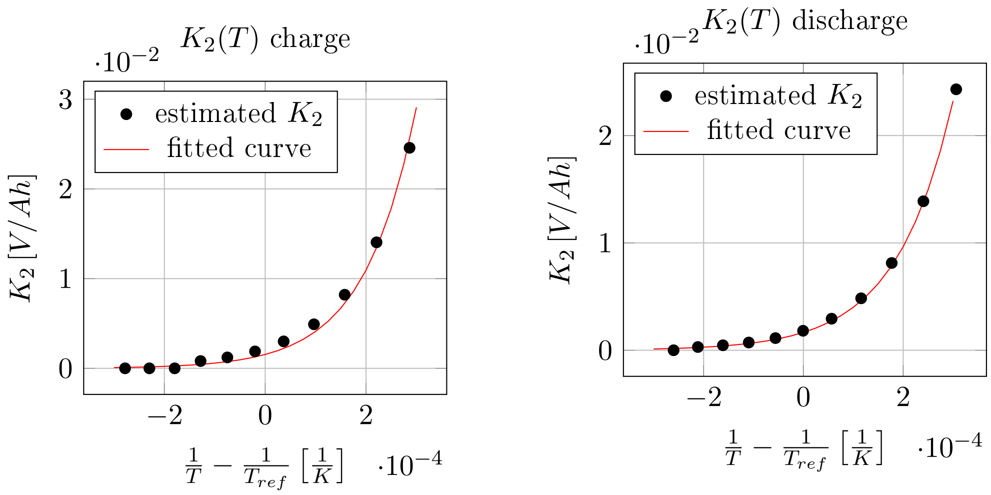
Figure 2.5: Estimation of the temperature dependency of Q



(a) The fitted thermal characteristics of parameter $K_1(T)$ from the charge data.

(b) The fitted thermal characteristics of parameter $K_1(T)$ from the discharge data.

Figure 2.6: Estimation of the temperature dependency of K_1



(a) The fitted thermal characteristics of parameter $K_2(T)$ from the charge data.

(b) The fitted thermal characteristics of parameter $K_2(T)$ from the discharge data.

Figure 2.7: Estimation of the temperature dependency of K_2

It can be seen from the estimated values that they are in good agreement with the nominal parameters of the investigated battery type, and coincide well with the parameters in the detailed dynamic battery model in *Simulink/Simscape/Electrical/Specialized Power Systems/Electric Drives/Extra Sources*.

The accuracy of the parameter estimation can be characterized by the *covariance matrix* of the estimation. In our results the elements of the covariance matrices are really small (with orders between 10^{-8} and 10^{-12}) in both charge and discharge cases. This means that the parameter estimation is very accurate. Note, that the experimental data were obtained *from the simulation of the model equations of the extended model with energy balance equation* and not from real measurements, therefore no external noise or modelling errors are included.

The uncertainty of the parameters can be characterized by the confidence region of the estimation. The confidence region of the estimated parameters can be approximated by the $1.05 \cdot W_{min}$ contour line of the cost function W. In order to analyze and illustrate the confidence regions, the parameters were analyzed in pairs in such a way that two of the parameters were fixed to their estimated values then the value of the cost function was computed while changing the other two parameter values around their estimated value. The parameter pairs were chosen to be E_0, Q and K_1, K_2. Some examples of the confidence regions in case of charge and discharge at different temperatures are illustrated on Figure 2.8 and Figure 2.9. The order of magnitude on the X and Y axes are the same in Figures 2.8a-2.8d and Figures 2.9a-2.9c respectively. In Figure 2.9d the axes are magnified for better visibility. Comparing the confidence regions at different temperatures and operating modes the following conclusions can be drawn:

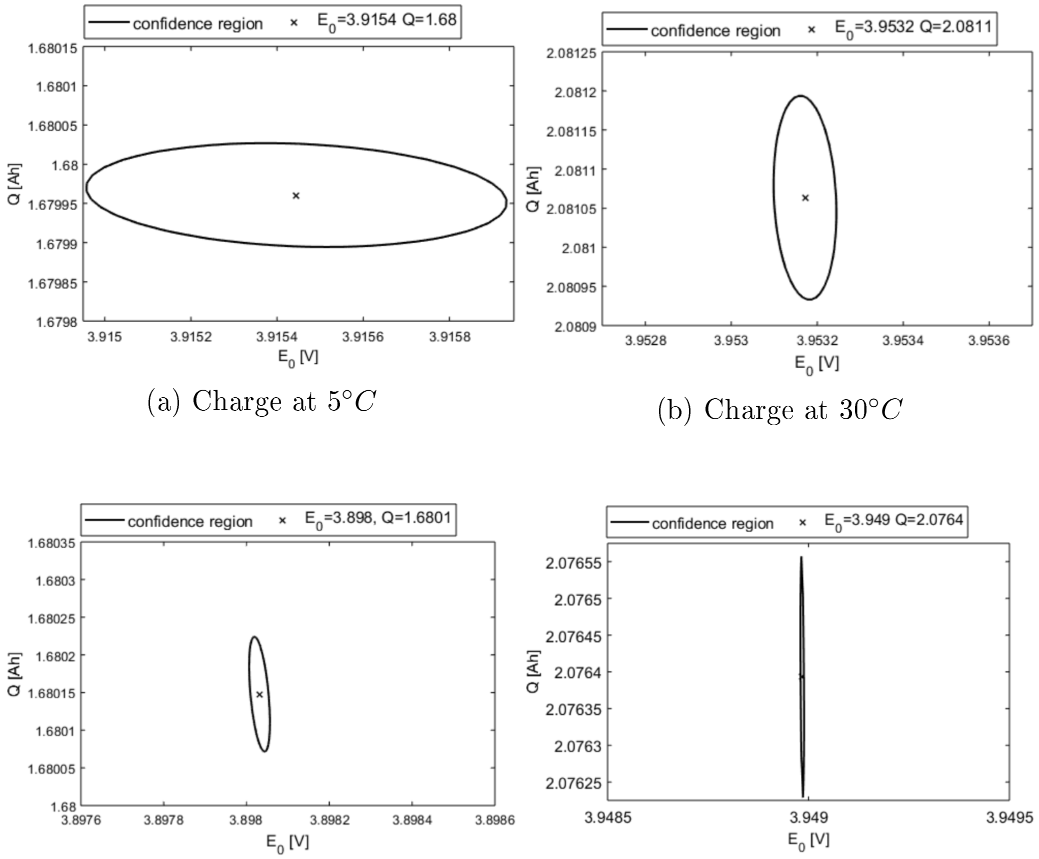
- In case of both charge-discharge, the confidence of Q increases while E_0 decreases as the temperature rises (see Figure 2.8a-2.8b, and Figure 2.8c-2.8d).

- E_0 and Q are uncorrelated because the axes of the ellipse are almost parallel with the X and Y axes.

- In case of charge, the confidence region of K_1, K_2 becomes smaller as the temperature rises (see Figure 2.9a and Figure 2.9b).

- A linear relationship between K_1 and K_2 can be assumed in case of discharge (see Figure 2.9c and Figure 2.9d).

Looking at Figures 2.4-2.5 it is apparent that the estimated values of E_0 and Q are more uncertain in case of charge. This phenomenon can be explained by the confidence regions depicted in Figure 2.8. It can be seen that the confidence region is wider in case of charge, hence the uncertainty of the parameters are greater. It can be also noticed that the shape of the confidence region changes with temperature. At low temperatures the confidence of the estimated Q is smaller than the confidence of E_0. On the contrary, at high temperatures the confidence of Q becomes greater while the confidence of E_0 decreases. That is why we can better estimate Q at low temperatures and E_0 at high temperatures. Additionally the estimates are results of nonlinear optimization which is affected by the initial values, stopping criteria, and the shape of the cost function.



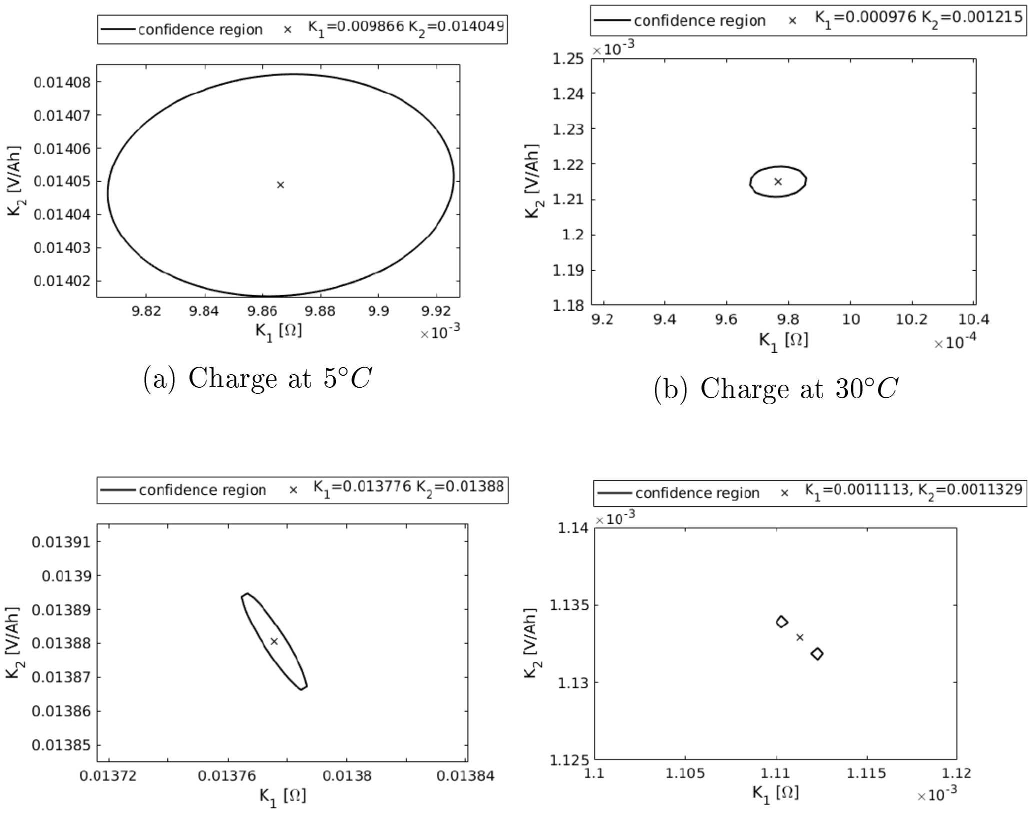
(a) Charge at 5°C

(b) Charge at 30°C

(c) Discharge at 5°C

(d) Discharge at 30°C

Figure 2.8: Confidence regions of the estimated parameters E_0, Q during charge/discharge at different temperatures. X axis: E_0, Y axis: Q, X axis range: $1 \cdot 10^{-3}$, Y axis range: $3.5 \cdot 10^{-4}$, $-$: confidence region, $\times$: estimated parameter value.



(a) Charge at 5°C

(b) Charge at 30°C

(c) Discharge at 5°C

(d) Discharge at 30°C

Figure 2.9: Confidence regions of the estimated parameters K_1, K_2 during charge/discharge at different temperatures. X axis: K_1, Y axis: K_2, X axis range: $1.25 \cdot 10^{-4}$, Y axis range: $7 \cdot 10^{-5}$, $-$: confidence region, $\times$: estimated parameter value.

2.5.2 Estimated temperature dependent parameters

Having estimated the battery parameters at different ambient temperatures, the temperature dependency of the parameters was estimated with the help of the Matlab Curve Fitting Toolbox [65]. Each parameter has two coefficients that describe the temperature dependency: the parameter value at the reference temperature and the temperature coefficient. The independent variables of the four different parameter estimation tasks are the following:

- $T - T_{ref}$, in case of $E_0(T)$;
- $T_a - T_{ref}$, in case of $Q(T_a)$;
- $\frac{1}{T} - \frac{1}{T_{ref}}$ in case of $K_1(T)$ and $K_2(T)$.

As it was mentioned in Section 2.4.2, the cell temperature T was substituted by the average surface temperature of the battery. The dependent variables are the estimated parameter values of the previous step that can be seen in Table 2.5 and Table 2.6.

The coefficients of the temperature dependency were estimated during both charge and discharge. The results of the estimation can be seen in Tables 2.7 and Table 2.8. The 95% confidence bounds shows the uncertainty of the estimated coefficients.

It can be seen that the estimated temperature dependency of E_0 and Q is close to the nominal values in both charge and discharge cases. The estimation of $Q|_{T_{ref}}$ and $\Delta Q/\Delta T$ is better in case of charge because the differences between the nominal and estimated parameter are smaller. However the estimation of the other parameters is better in case of discharge.

Table 2.7: Estimated parameters of the temperature dependency of the battery parameters during charge.

Parameter	Nominal value	Estimated value	95% confidence bounds	Unit
$E_0\|_{T_{ref}}$	3.9388	3.943	(3.94, 3.946)	V
$\partial E/\partial T$	$2.0 \cdot 10^{-3}$	$1.518 \cdot 10^{-3}$	($1.314 \cdot 10^{-3}$, $1.723 \cdot 10^{-3}$)	V/K
$Q\|_{T_{ref}}$	2.0	2.001	(2.0, 2.001)	Ah
$\Delta Q/\Delta T$	$1.6 \cdot 10^{-2}$	$1.605 \cdot 10^{-2}$	($1.601 \cdot 10^{-2}$, $1.610 \cdot 10^{-2}$)	Ah/K
$K_1\|_{T_{ref}}$	$1.8 \cdot 10^{-3}$	$2.735 \cdot 10^{-3}$	($1.866 \cdot 10^{-3}$, $3.604 \cdot 10^{-3}$)	Ω
α_1	8415	5989	(4684, 7294)	K
$K_2\|_{T_{ref}}$	$1.8 \cdot 10^{-3}$	$1.545 \cdot 10^{-3}$	($1.866 \cdot 10^{-3}$, $1.987 \cdot 10^{-3}$)	V/Ah
α_2	8415	9785	(8706, 10860)	K

Table 2.8: Estimated parameters of the temperature dependency of the battery parameters during discharge.

Parameter	Nominal value	Estimated value	95% confidence bounds	Unit
$E_0\|_{T_{ref}}$	3.9388	3.939	(3.938, 3.939)	V
$\partial E/\partial T$	$2.0 \cdot 10^{-3}$	$2.025 \cdot 10^{-3}$	($2.009{\cdot}10^{-3}$, $2.041{\cdot}10^{-3}$)	V/K
$Q\|_{T_{ref}}$	2.0	1.995	(1.993, 1.997)	Ah
$\Delta Q/\Delta T$	$1.6 \cdot 10^{-2}$	$1.568 \cdot 10^{-2}$	($1.554{\cdot}10^{-2}$, $1.581 \cdot 10^{-2}$)	Ah/K
$K_1\|_{T_{ref}}$	$1.8 \cdot 10^{-3}$	$1.588 \cdot 10^{-3}$	($1.418{\cdot}10^{-3}$, $1.757{\cdot}10^{-3}$)	Ω
α_1	8415	8908	(8528, 9289)	K
$K_2\|_{T_{ref}}$	$1.8 \cdot 10^{-3}$	$1.661 \cdot 10^{-3}$	($1.542{\cdot}10^{-3}$, $1.781{\cdot}10^{-3}$)	V/Ah
α_2	8415	8793	(8538, 9048)	K

The fitted curves of the temperature dependency can be seen in Figures 2.4-2.7 with red line.

The goodness of the fit was characterized by the r^2 value that is computed by:

$$r^2 = 1 - \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (\hat{y}_i - \bar{y})^2}$$

where $\hat{y}$ is the measured data, y is the model predicted value, and $\bar{y}$ is the mean of the measured data. The results can be seen in Table 2.9. It can be seen that the curve fitting is a bit more accurate in case of discharge, except for Q.

Table 2.9: Goodness of curve fitting characterized by the r^2 value.

Parameter	E_0	Q	K_1	K_2
r^2 (charge)	0.9691	1	0.9656	0.9925
r^2 (discharge)	0.9999	0.9999	0.9995	0.9988

2.6 Discussion and future work

In practice the presented method can be used in battery management systems for example in automotive applications. The estimation of battery age and remaining life, the available capacity and state of charge at different temperatures are possible fields of use.

Further research directions include the use of this parameter estimation method for determining the state of health of the battery, and to estimate the temperature dependent state of charge during its life cycle. This is possible through a suitable experiment policy that estimates the battery capacity from well chosen charging operations under different thermal conditions. Therefore, extensive climate chamber experiments will be performed to validate the results the presented work.

Simulations will also be carried out to analyze the effect of temperature on the aging of the battery. Matlab Simulink provides a battery model with both temperature and aging effects, which can help to develop a proper diagnostic method. Simulations with different battery parameters can also be used to demonstrate the capabilities of the diagnostic method.

The effect of rapid charge/discharge can also be part of further research. In that case the temperature of the battery increases significantly, therefore the assumptions about the battery temperature are no longer valid. The high variation of the cell temperature may affect the estimated values of the parameters, therefore the results will not be reliable. In that case the parameter estimation method should be modified.

2.7 Summary

An optimization based lithium-ion battery parameter estimation method has been proposed in this chapter that is capable of describing the thermal behaviour of batteries. The base of the method is a nonlinear charge and discharge model which describes the temperature dependency as a parametric function of temperature as an external variable.

Parameter sensitivity analysis has been carried out on the model to find the parameters to be estimated, that are the electrode potential, the battery capacity, and two polarization constants. The model output was found to be non-sensitive to the internal resistance, thus it was not estimated.

The proposed parameter estimation method contains two steps. At first the parameters E_0, Q, K_1, K_2 are estimated from measured data of charging/discharging at different constant ambient temperatures. In the second step the temperature coefficients of these parameters are estimated.

The proposed parameter estimation method is verified by a set of simulation experiments on an electro-thermal battery model capable of describing the energy balance (i.e. the thermal behaviour) of the battery. The temperature dependent parameter characteristics obtained by the proposed method can be used as a way of determining the key battery parameters at a given temperature. The *novelty of the method* is that the temperature dependent parameter characteristics can be estimated from charging profiles by the proposed method, that enables to use it in practice for determining the key battery parameters at a given temperature.

Chapter 3

Model based diagnosis of electrical networks

The demand of electrical energy is continuously increasing all over the world. Electrical energy is generated by power plants or renewable energy sources and is transmitted to the consumers through distribution stations and power lines. During the transmission there are technical losses, which reduce the efficiency of the power delivery. They come from dissipation in the conductors, transmission lines, substation transformers and magnetic losses in transformers. The amount of technical losses is about 20% of the total transmitted energy.

Besides the technical losses there may be non-technical losses (NTLs) too. These are unnecessary losses which are not expected and cannot be planned ahead. The NTLs are usually related to energy theft and fraudulent consumer behaviour. Electricity theft has been a widespread and major issue for many years and various techniques of energy theft are present from unregistered users to hacking the metering device [66]. NTLs caused by companies in the US were estimated between 0.5% and 3.5% of annual gross revenues. The NTLs may be more than 15% of the generated power in some other countries [67].

Following this unwanted phenomena, several methods for non-technical loss detection appeared in the literature. It can be stated that there is no golden rule for detecting electricity theft but there are several different approaches [68]. The papers [69] and [70] give very good review of the most frequently used methods in the field.

The majority of the solutions available in the literature are based on the analysis of consumption data using some statistical or machine learning methods. For example, the authors of [71] use a linear regression based procedure that not only detects electricity theft but the defects of the smart meters can also be detected. A probabilistic neural network based classification approach is presented in [72], where the Levenberg-Marquardt method is used for training the network. A support vector machine based solution is given in the work [73], where a parallel computing architecture was proposed in order to enhance computation.

Another direction of non-technical loss detection is based on the network topology, such network-oriented methods use measurement data from grid sensors and smart meters [74]. In [75] authors proposed a Kalman filter state estimator to find currents and biases in a microgrid network. The currents and biases are estimated separately with two different filters. If the estimated bias of a customer is larger than the predefined threshold then this user committed fraud. The authors of [76] suggest a probabilistic power flow approach for NTL detection. The output of the algorithm is a probability distribution of NTLs in the subnetwork. A combination of the above mentioned statistical and network oriented approaches has appeared in a recent paper [77], where the past consumption data were used as nominal data.

Almost all of the above mentioned approaches of non-technical loss detection and localization belong to the class of model-based fault detection and isolation, see e.g. in [8]. In this case the model of electrical networks are needed, where both of their dynamic and steady-state models are used for diagnostic purposes of different kind.

In practical cases, however, one needs to consider the effect of uncertainties and to cope with the situation when relatively rare measurements are available with possible measurement errors, and the available measurements are not uniformly placed over the network. The uncertainties can be handled either by using stochastic models or by considering interval-valued variables in the model. In my previous work a Colored Petri Net model, that is a dynamic discrete event model, of electrical networks was developed [78]. Here interval-valued variables turned out to be good for modelling the parameter uncertainties. Moreover it was found, that a static model may be enough for diagnosing non-technical losses based on measured data in the simplest case.

Decomposition offers the potential to reduce the complexity of model-based optimization, prediction, control and diagnosis by accounting for the structure and sparsity of the describing model. Motivated by the fact that the energy systems, including electrical networks, are most often large scale complicated systems, a rich and powerful collection of decomposition methods are available for different purposes. The recent paper [79] proposes a decomposition of electric power network based on its algebraic model for planning purposes. Intelligent partitioning methods, that use heuristics besides of traditional mathematical methods are also available for smart grids [80]. Powerful graph-based decomposition methods for model structures are also reported recently, see e.g. [81], [82].

The aim of the work described in this Chapter was to develop a diagnostic method for non-technical loss detection and localization. The conclusion of our previous work, is that a static linear model is enough for the diagnosis. It was used in this work too, but in a different representation form. The approach followed belongs to the network-oriented class and it is based on analyzing the differences between the measured and model-predicted "nominal" voltages, i.e. an analytical redundancy based method is used for utilizing the capabilities of smart meters. The novelty of the proposed method is that structural decomposition of the network is performed before the diagnosis. Utilizing the network structure, the diagnosis can be performed in parallel on smaller subnetworks. A novel fault isolation method is proposed that computes the illegal currents of successive nodes and analyzes the change in their value. The proposed diagnostic algorithm takes into account the uncertainty in the model parameters together with the measurement uncertainties to make the approach applicable in real-world cases. Another advantage of the diagnostic method is that, it is able to localize multiple faults in one step and approximate the fault magnitude.

3.1 Electrical network models, basic assumptions

Electrical networks are composed of several electrical components that are connected to each other. They can be modelled by each of the general model types (static/dynamic, linear/nonlinear, discrete/continuous, deterministic/ non-deterministic) with different capabilities. The classical models of electrical networks are in the form of linear and/or nonlinear ordinary differential equations in the dynamic, and of linear and/or nonlinear algebraic equations in the static case (see e.g. in [83]–[85]). The effect of the model type on the performance of the fault detection and isolation methods is also thoroughly analyzed in the literature, see e.g. [86]–[89].

As the electrical network is a set of connected elements, graph theory can be also used to create electrical network models. Graph models have the advantage that they provide a graphical representation of the network structure and graph algorithms can be applied to them [90]. In this thesis a simple static linear model of the electrical network with directed graph representation is used. The basic notions and assumptions about the model applied for detection and isolation of non-technical losses are described in this section.

3.1.1 The model of the electrical network

The modelling approach of the electrical network used for model-based diagnosis is described in this section by giving the modelling assumptions together with the model elements and their connections. The applied notations are collected in Section 3.1.4.

Modelling assumptions

The role of electrical grids is to transfer the generated energy to the consumers. The transmission is done in several stages: from power supplies to distribution stations and then to customers. In this work the *low voltage grid of one transformer area* is considered with a radius of 1-1.5 km. A typical example of such transformer area is a transformer station, which supplies energy to the customers in a few streets of a town.

During the work the following basic assumptions are made:

1. The model covers one transformer area that is composed of transformers, loads (customers) and transmission lines.

2. A simple static linear model of the electrical network is used. The known model parameters are the resistances of the transmission lines.

3. The current and the voltage of the transformer is measured.

4. Smart meters are installed at every load to measure the current, voltage and the consumed power.

5. There may be one or more loads in the network that have fraud meters (for more information about fraud meters see Section 3.1.2).

6. The resistances of the wires between the loads and the junction points are usually much smaller than the resistance of the main transmission line. Therefore resistances of the wires between the loads and the junction points are neglected.

7. The model uncertainties come from the resistances of the wires in the main transmission lines.

8. The investigated grid is one point grounded.

Model elements and their connections

The investigated electrical networks are composed of the following basic elements:

- *Feeders* represent the transformers in the network.
- *Loads* represent the registered users (consumers) in the network.
- *Wires* represent the transmission lines between the feeders and the loads.

The network is represented by a directed graph. The nodes of the graph are the feeders, loads and the junction points of the wires. The edges between the nodes represent the wires of the network. The edges are always between loads and junction points, junction points and loads or between two junction points. There are no direct edges between two loads or between loads and feeders. The direction of the edges represent the direction of the current flow.

The nodes and edges have different attributes attached to them:

- Feeder: measured current and voltage of the feeder (I_F, U_F);
- Load: measured current and voltage of the load (I_i, U_i);
- Junction point: computed voltage of the node (U_j);
- Wire: resistance of the wire, computed current of the wire (R_{w_i}, I_{w_i}).

The *minimal required data for computing all of the current and voltage values of the network* are the currents and voltages of the feeders, the currents of loads and the resistances of the wires. The resistances of the wires are not measured but they can be computed knowing the diameter, length and resistivity, which are given by the utility provider. The problem is that these values are temperature dependent:

$$R = R_{ref}(1 + \alpha(T - T_{ref})), \tag{3.1}$$

where R is the resistance of the wire at temperature T, R_{ref} is the resistance of the wire at reference temperature T_{ref} and α is the temperature coefficient for the transmission line material. The temperature coefficient for copper is 0.004041, for aluminium is 0.004308 at 20 °C, see e.g. [91]. Because the temperature of the transmission line is varying the computed resistance depends on the temperature, too. To handle this uncertainty caused by the temperature dependency, a *percentage based parameter uncertainty of the resistances* was introduced. The typical value of this parameter uncertainty is about ± 2 % if the error of temperature estimation of the transmission line is ±5°C.

3.1.2 Technical and non-technical losses

There are several types of non-technical losses that usually originated from improper usage. According to [70], the non-technical losses can be classified as follows.

- Before meter: tapping of the transmission lines;
- Meter: manipulating the meter in order to lower the registered consumption (e.g. the meter is bypassed, disconnected or reversed);
- Billing: losses due to the inaccurate operation of the billing system, unpaid bills, etc.

In this work it is assumed that the examined non-technical losses belong to the second category (i.e. the loss is at the meter) including the following three types of NTLs:

- fraud with the meter (reversing or hacking the meter hardware, software or calibration);
- bypassing the meter;
- inaccurate measurement.

The users in the network may have a trusty or a fraud meter. In this paper the terminology of legal and illegal loads are used to distinguish them.

- The *legal loads* have a registered and certified meter. This meter shows the real energy consumption of the user.
- The *illegal loads* have a fraud meter, i.e. the meter is reversed, bypassed, or manipulated in such a way that the measured energy consumption is less than the real consumption. The non-technical losses in the network are resulted by the illegal loads.
- The *illegal current* is the extra current of the load that is not measured.

Note that all users are registered ones and unregistered users are not considered in this work.

3.1.3 Measurements

Measurement data is essential for the diagnosis of the network. In this work the measured points are located at the loads and the feeders of the network. The conventional electricity meters measure only the electrical energy consumption. However, nowadays the *smart meters* become more common both in residential and industrial environment. Smart meters can measure real-time consumption data, active and reactive power and the average voltage and current of the customers over the measurement time intervals. The data are recorded at every 15 minutes and can be sent automatically to the electricity supplier. This sampling time is much greater than the time period of the transient events, which are in the milliseconds range. Therefore a static model is sufficient enough to compute the network variables. In the proposed diagnostic method *it is assumed that every load and feeder is equipped with a smart meter, which measures the current and voltage of the load.*

Any measurement always has a measurement error, which cannot be ignored in real applications. The measurement error of smart meters usually falls between ± 0.2% and ± 2%. Typical values of the measurement error of a smart meter are:

- Voltage: ± 0.2% (50-830 VAC)
- Current: ± 0.2 % (0.1-1.1 I nominal)
- Power: ± 0.5% Full Scale

For more details about smart meter accuracy see standard EN 62053:2003.

3.1.4 Notations

Variables and parameters:		Indices:	
U	nominal voltage	$_{F_i}$	ith feeder
$\tilde{U}$	measured voltage	$_i$	ith load
ΔU	voltage difference	$_{w_i}$	ith wire
I	nominal current	$_{ki}$	kth branch ith load
$\tilde{I}$	measured current	$_{w_{ki}}$	kth branch ith wire
R	resistance	$_{ill_i}$	ith illegal load
ε_I	current measurement error		
ε_U	voltage measurement error		
ε_R	uncertainty of resistance		

3.2 Decomposition of electrical networks

In this section the different structural layouts of electrical networks and their structural decompositions are introduced.

The decomposition of the network is a crucial part of the proposed diagnosis method. The objective is to decompose the network into subnetworks with basic one feeder layouts. During the decomposition it is determined which loads belong to the same subnetwork then the NTLs are detected and localized within these subnetworks. The advantage of the decomposition is that the diagnosis can be performed in parallel on the subnetworks, that increase the efficiency of the method.

The four basic layouts are introduced in Section 3.2.1. Then the decomposition algorithms for one and two feeder networks are introduced in Section 3.2.2. Finally the decomposition methods are illustrated on simple examples in Section 3.2.3. The notations summarized in Section 3.1.4 are used in the description.

3.2.1 Basic structures of electrical networks

There are four basic structural layouts of low voltage electrical networks that are presented below.

One feeder layout

The simplest structure is the *one feeder layout.* It is composed of one feeder and several loads connected to the feeder with one common transmission line. The loads are supplied with electrical energy from the feeder through that common transmission line. The structure of the network can be seen in Figure 3.1. The resistances and the currents of the ith wire of the transmission line are denoted by R_{w_i} and I_{w_i}, while the jth load is characterized by its current I_j. The voltage and current of the feeder is denoted by U_F and I_F.

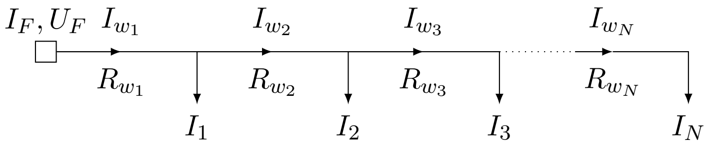


Figure 3.1: One feeder layout

Radial feeder layout

The *radial feeder layout* is a bit more difficult than the one feeder layout. It has one feeding point too, but there are branches on the main transmission line with one or more loads on the branches. The loads are placed in a line similarly to the one feeder layout. The directions of the edges are clearly defined as the current flows from the feeder to the loads. The schematic drawing of a radial feeder network can be seen in Figure 3.2, where the branches are indexed starting from the feeder, and the ith wire of the transmission line of branch k has the index identifier w_{ki}, e.g. its resistance is $R_{w_{ki}}$.

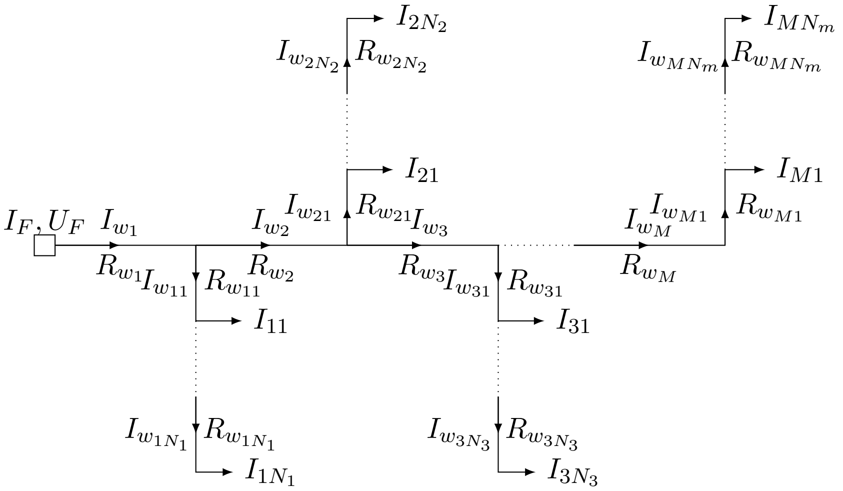


Figure 3.2: Radial feeder layout

Two feeder layout

Sometimes one feeder is not enough to provide enough energy to a given area or the energy supplier would like to build one transmission line with redundant feeder point. In that case a new feeder can be added to the network to provide the required energy. This structure is called the *two feeder layout.* In this layout there are loads that are fed by either the first or the second feeder and there may be one load that is fed by both feeders. The directions of the edges are not clearly defined because the current direction depends on the magnitude of the load currents. Therefore the directions of the edges can be determined in parallel with the computation of the network variables. The simplest form of this layout is the extension of the one feeder layout that can be seen in Figure 3.3. Two feeders give safety service for the customers because there is no dropout with one cut-off point. Of course the radial network can also be extended into a two feeder layout form by placing a second feeder to the other end of the main transmission line resulting in a *two feeder radial layout.*

Loop feeder layout

The last basic type of network structures is the *loop feeder layout.* In this layout the two ends of the transmission line are connected to the same feeder. This kind of network can be seen as a two feeder network where the two feeders are identical. The simplest structure of the loop feeder layout can be seen in Figure 3.4. There may be branches in the loop feeder layout too.

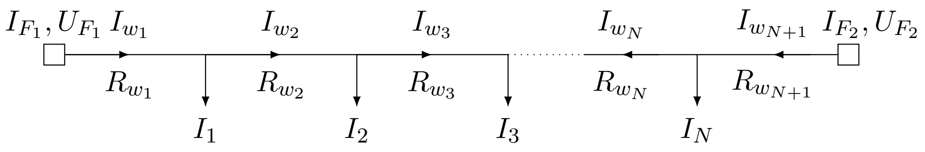


Figure 3.3: Two feeder layout

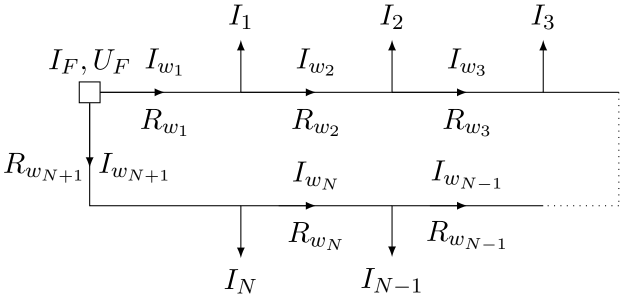


Figure 3.4: Loop feeder layout

3.2.2 The decomposition method

The diagnosis of a complex system is always a difficult problem. The faults can be in different parts of the system and the faults may have effects on other subsystems too. Therefore it is always profitable to perform the diagnosis in a distributed way using smaller and simpler subsystems.

Fortunately, the electrical networks with all of the layouts introduced in Section 3.2.1 can be decomposed into smaller subsystems, with the structure of one feeder layout.

Decomposition of the one feeder radial network

The radial network is different from the one feeder network in that there are branches in the transmission line. The branches have one or more loads connected to them or even new branches may be present. The decomposition of a radial network is based on cutting off the branches of the network until there are no branches in the subsystems. The resulted subsystems are all networks with one feeder layout. In the directed graph representation of a radial network there are three types of nodes:

- The node with in-degree of 0 is the feeder.
- The nodes with out-degree of 0 are the loads.
- The nodes with out-degree more than 1 are the junction points.

The decomposed network has two types of subnetworks:

- *Multiple load subnetworks* where more than one loads belong to a subnetwork.

- *Single load subnetworks* where only one load belongs to a subnetwork.

Algorithm 1 Decomposition of the radial network

```
$G = (V, A)$
function DecompRadial($G$)
    $L = \{v \in V : deg^+(v) = 0\}$                         // load nodes
    $X = \{v \in V : deg^-(v) > 1\}$                         // junction nodes
    $J = \emptyset$                  // junctions whose successors are also junctions
    $J_S = \emptyset$                                   // successors of nodes in $J$
    for all $j \in J$ do
        $N = suc(j)$                              // suc(j): successors of $j$ in $G$
        if $N \subseteq X$ then
            $J = J \cup \{j\}$
            $J_S = J_S \cup N$
        end if
    end for
    $A' = A \setminus (J \times J_S)$                  // remove edges between $J$ and $J_S$
    $G' = (V, A')$                                    // preliminary decomposed graph
    $C = \{C_1 = (V_{C_1}, A_{C_1}), \ldots, C_n = (V_{C_n}, A_{C_n})\}$
                                              // set of connected components of $G'$
    $i = 1$
    for $k = 1$ to $n$ do
        if $\exists l \in V_{C_k} \cap L$ then               // components with at least one load
            if $\exists x \in V_{C_k} \cap J$ then   // components with at least one node in $J$
                for $\forall l \in V_{C_k} \cap L$ do        // separating single load subnetworks
                    $V_k = \{l, pred(l)\}$                // pred(l): predecessors of $l$
                    $A_k = (pred(l), l)$
                    $S_i = (V_k, A_k)$                  // subgraph of $G'$ with nodes $l$ and $p$
                    $i = i + 1$
                end for
            else                                          // multiple load subnetworks
                $S_i = C_k$                         // subgraph of $G'$ with nodes in $C_k$
                $i = i + 1$
            end if
        end if
    end for
return $S$
end function
```

The decomposition method is composed of two main steps:

1. Cut the graph along junction points, whose successors are also junction points to get a preliminary decomposition. Determine the connected components of the graph.

2. Analyze the connected components and select the ones that have loads in them. These components are either real subnetworks or can be decomposed into single load subnetworks.

The *decomposition algorithm* of the networks with radial layout can be seen in Algorithm 1.

At first the nodes of the graph are classified as loads (L) or junctions (X). (Feeders are not considered because they are irrelevant from the decomposition point of view.) Then two sets are initialized as empty:

- J is the vector of those junction nodes, whose successors are also junctions.

- J_S is the vector of successors of J.

Having initiated these sets the appropriate junction nodes are added to the J and J_S. The preliminary decomposition step is based on the idea that the edges between nodes in J and J_S can be deleted from the graph because they are connecting different subnetworks. After this step the graph falls apart into disconnected subgraphs, which are connected in themselves. The set of these components are denoted by C and the connected components are denoted by $C_k = (V_{C_k}, A_{C_k})$.

In the second part of the decomposition the C_k components are categorized according to their nodes. At first, the subgraphs that have loads between their nodes are selected. Then these subgraphs are further analyzed: if the subgraph has at least one node in the set J, it means that this subgraph is not fully decomposed. However it may occur only in that case when the subgraph is composed of single load subnetworks. Therefore this subgraph is decomposed in such a way that the loads (l) and their predecessors ($pred(l)$) are added to the ith subnetwork S_i. In the other case, when the component does not contain any nodes in J, then it is already a complete subnetwork and no further decomposition is needed.

Decomposition of the two feeder network

The decomposition of the two feeder network is different from the one feeder case, because the directions of the edges are initially not known. The two feeder network operates in such a way, that half of the network is fed by the first feeder and the other half is fed by the second feeder. The point where the two parts of the network meet can be determined during the computation of the wire currents. Therefore the computation of the network variables, the determination of the edge directions and the decomposition are done in parallel in case of the two feeder network. Knowing the directions of the edges the decomposition is quite straightforward. The steps of the decomposition are the following.

Algorithm 2 Decomposition of the basic two feeder network

```
$G = (V, A)$
**function** DecompTwoFeeder($G$)
    $F = \{v \in V : deg(v) = 1 \wedge I = \emptyset\} = \{F_1, F_2\}$    // feeder nodes
    $X = \{v \in V : deg(v) > 1\} = \{X_1, \ldots, X_{N_L}\}$    // junction nodes
    $L = \{v \in V : deg(v) = 1 \wedge I \neq \emptyset\} = \{L_1, \ldots, L_{N_L}\}$    // load nodes
    $V_1 = \{F_1\}, A_1 = \emptyset$
    $V_2 = \{F_2\}, A_2 = \emptyset$
    $I_{F_1} = I_{w_1} = \sum_{i=1}^{N_L} I_i - \left(\sum_{i=1}^{N_L}\left(I_i \sum_{j=1}^{i} R_{w_j}\right)\right) / \sum_{j=1}^{N_W} R_{w_j}$
                                                    // current of the first wire
    $i = 1, X_0 = F_1$
    **while** $I_{w_i} > 0$ **do**
        $V_1 = V_1 \cup X_i, L_i$
        $A_1 = A_1 \cup \{(X_{i-1}, X_i), (X_i, L_i)\}$
        $I_{w_{i+1}} = I_{w_i} - I_i$
        $i = i + 1$
    **end while**
    **if** $I_{w_i} = 0$ **then**
        $V_2 = V \setminus V_1$
        $A_2 = (A \setminus A_1) \setminus (X_{i-1}, X_i)$
        $S_1 = (V_1, A_1)$
        $S_2 = (V_2, A_2)$
    **else**$I_{w_i} < 0$
        $I'_{i-1} = I_{i-1} + I_{w_i},\ I''_{i-1} = -I_{w_i}$
        $V_1 = (V_1 \setminus \{L_{i-1}\}) \cup \{L'_{i-1}\},\ A_1 = A_1 \cup \{(X_{i-1}, L'_{i-1})\}$
        $V_2 = (V \setminus V_1 \setminus \{L_{i-1}\}) \cup \{X_{i-1}, L''_{i-1}\},$
        $A_2 = \{(F_2, X_{N_L}), (X_{N_L}, X_{N_L-1}), (X_{N_L}, L_{N_L}), \ldots,$
        $(X_i, X_{i-1}), , (X_{i-1}, L''_{i-1})\}$
        $S_1 = (V_1, A_1)$
        $S_2 = (V_2, A_2)$
    **end if**
**return** $S_1, S_2$
**end function**
```

1. The feeders are those nodes in the graph whose degree is 1 but the current of the node is not known. The currents of the feeders can be computed knowing the currents of the loads and the resistances of the wires using the equations:

$$I_{F_1} + I_{F_2} = \sum_{i=1}^{N_L} I_i \tag{3.2}$$

$$I_{F_2} \sum_{j=1}^{N_W} R_{w_j} = \sum_{i=1}^{N_L} (I_i \sum_{j=1}^{i} R_{w_j}) \tag{3.3}$$

   where N_L is the number of loads and $N_W = N_L + 1$ is the number of wires in the network.

2. Starting from one of the feeders, the currents of the loads are subtracted one by one from the current of the transmission line. The current of the ith wire is computed with the equation below.

$$I_{w_i} = I_{w_{i-1}} - I_{i-1}$$

3. At some point a load (L_i) is reached where the current becomes less or equal to zero. If the current is equal to zero than it means that the feeder cannot fed more loads after this load. Therefore the network should be cut in half exactly after the load L_i.

4. If the load current at L_i is less than zero than it means that this load is fed by both feeders, and the remaining (negative) current is supplied by the other feeder. In that case the load is also divided into two loads L_i' and L_i'' with proper parts of the feeder currents. Then the network is cut in half in such a way that L_i' belongs to the first subnetwork and L_i'' belongs to the second subnetwork.

5. The two resulting subnetworks are composed of one feeder and the loads that are fed by that feeder.

If the two feeder network has radial layout then first it needed to be converted into a basic two feeder layout network (such as in Figure 3.3) by cutting out its branches. To do this the loads that are in a branch of the transmission line are merged together.

In order to determine which branches should be merged the shortest path between the two feeders need to be found. Then the loads of the branches whose start node is in the path can be merged together. This *merged load substitutes the whole branch*. The current of this load is the sum of the current of the loads in the branch. The resulting network is a basic two feeder layout but the loads are the merged loads of the branches. After that the currents of the feeders can be computed and the two subnetworks can be created using the method described above.

The algorithm of decomposing basic two feeder networks are given in Algorithm 2.

3.2.3 Simple examples

In this section the decomposition methods of one and two feeder networks are illustrated on simple examples.

Radial one feeder network

The first example is the decomposition of a radial one feeder network. The directed graph of the example network can be seen in Figure 3.5a. The network has 10 loads with node labels $L1, \dots, L10$. Node with label $F1$ is the feeder. Nodes labelled by X are the junction points.

Algorithm 1 is used in the following steps.

1. The algorithm first determines the vector of loads and junction points. In this case $L = [L1, L2, L3, L4, L5, L6, L7, L8, L9, L10]$ and $X = [X1, X2, X3, X4, X5, X6, X7, X8, X9]$.

2. Then the J and J_S sets are created with $J = \{X1, X2, X6\}$ and $J_S = \{X2, X9, X3, X5, X7, X8\}$. The nodes in J are denoted by squares in Figure 3.5b and the nodes in J_S are denoted by big blue circles. Note that $X2$ is in both sets that is why it is denoted by a blue square.

3. In the next step the edges that start in in J and end J_S are removed from the graph, namely the edges between nodes $(X1, X2), (X1, X9)$, $(X2, X3), (X2, X5), (X6, X7)$ and $(X6, X8)$. The removed edges are denoted by dashed lines in Figure 3.5b.

4. By removing the dashed edges the new G' graph falls apart into some connected graphs. The ith element of the set C contains the nodes of one connected component of G'. In this example there are 7 components after removing the specified edges:
$C_1 = (V_{C_1}, A_{C_1}),\ V_{C_1} = \{F1, X1\},\ A_{C_1} = \{(F1, X1)\}$
$C_2 = (V_{C_2}, A_{C_2}),\ V_{C_2} = \{X2\},\ A_{C_2} = \emptyset$
$C_3 = (V_{C_3}, A_{C_3}),\ V_{C_3} = \{X3, L3, X4, L2, L3\}$,
$A_{C_3} = \{(X3, L1), (X3, X4), (X4, L2), (X4, L3)\}$
$C_4 = (V_{C_4}, A_{C_4}),\ V_{C_4} = \{X5, L4, X6\},\ A_{C_4} = \{(X5, L4), (X5, X6)\}$
$C_5 = (V_{C_5}, A_{C_5}),\ V_{C_5} = \{X7, L5, L6\},\ A_{C_5} = \{(X7, L5), (X7, L6)\}$
$C_6 = (V_{C_6}, A_{C_6},\ V_{C_6} = X8, L7, L8),\ A_{C_6} = \{(X8, L7), (X8, L8)\}$
$C_7 = (V_{C_7}, A_{C_7}),\ V_{C_7} = \{X9, L9, L10\},\ A_{C_7} = \{(X9, L9), (X9, L10)\}$

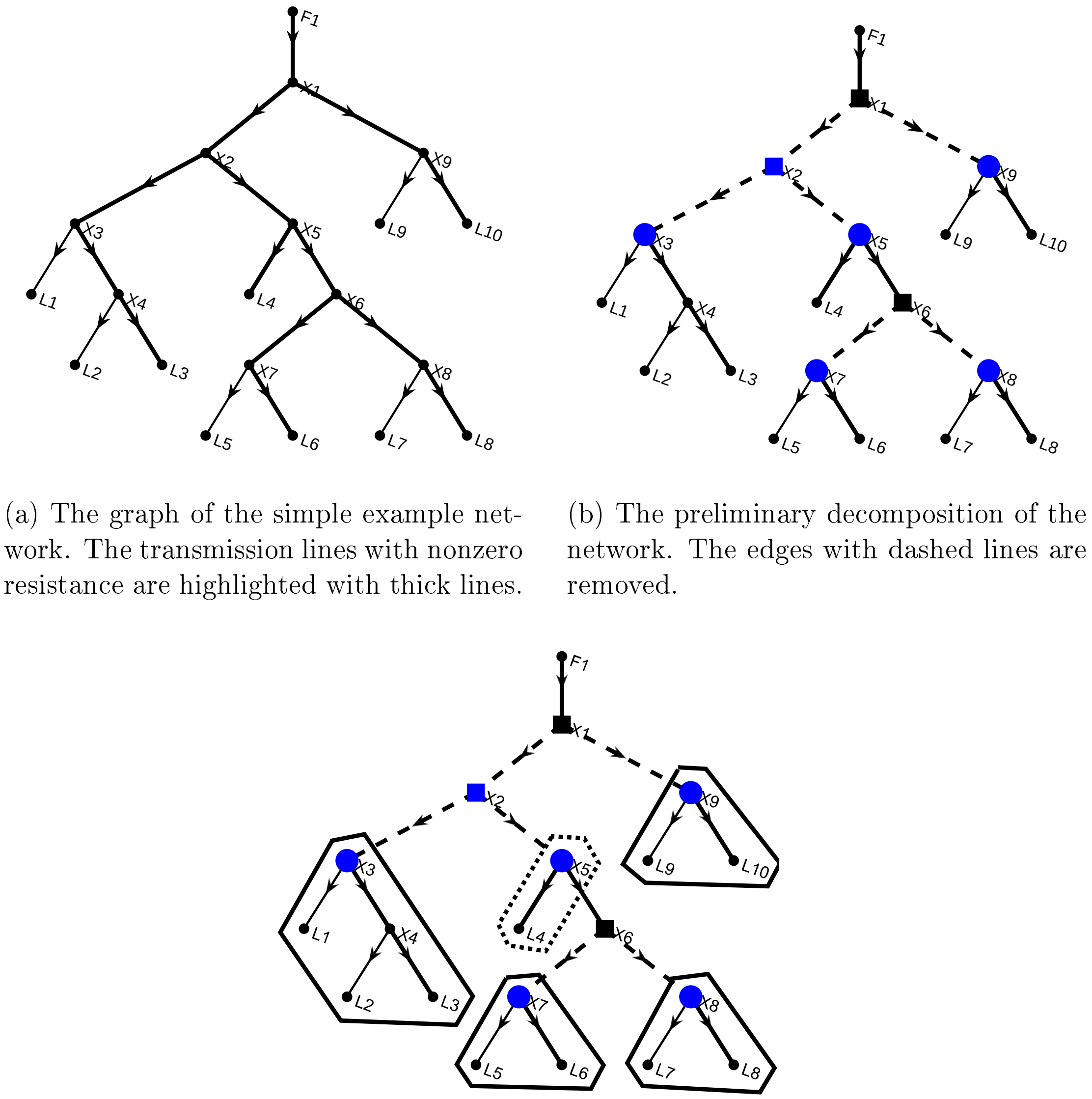
5. Then we go through these components and examine whether they contain any loads. The components that do not contain any load, i.e. C_1 and C_2, are not considered in the final decomposition.

6. The other components are examined whether they can be divided into smaller components. C_3 is already a good one feeder layout network, so the first subnetwork S_1 is equal to C_3. C_4 contains an extra junction node (X6) that needs to be eliminated from the network to get a one feeder layout. After that the second subnetwork is $S_2 = (V_2, A_2)$, where $V_2 = \{X5, L4\}$ and $A_2 = \{(X5, L4)\}$. The components C_5, C_6 and C_7 are similar to C_3 therefore the further subnetworks are $S_3 = C_5,\ S_4 = C_6$ and $S_5 = C_7$.

In conclusion we get the following subnetworks:

- Multiple load subnetworks: S_1, S_3, S_4, S_5
- Single load subnetwork: S_2

The decomposed network with its subnetworks is displayed in Figure 3.5c. The single load subnetwork is framed by dotted line and the multiple load subnetworks are framed by solid line.



(a) The graph of the simple example network. The transmission lines with nonzero resistance are highlighted with thick lines.

(b) The preliminary decomposition of the network. The edges with dashed lines are removed.

(c) The final decomposition of the network. The multiple load and single load subnetworks are framed with solid and dotted lines respectively.

Figure 3.5: The radial network of the simple example and its structural decomposition.

Two feeder networks

The decomposition of the two feeder layout is illustrated on two examples.

Basic two feeder network The first example is a basic two feeder network without any branches in the transmission line. The network of this example can be seen in Figure 3.6a. Note that the directions of the edges can also be determined during the decomposition. The nodes with labels $L1, L2, L3, L4$ and $L5$ are the loads and their nominal currents are $I_1 = 3A, I_2 = 7A, I_3 = 5A, I_4 = 4A, I_5 = 8A$ respectively. The resistances of the wires are $R_{w_1} = 0.1\Omega, R_{w_2} = 0.05\Omega, R_{w_3} = 0.07\Omega, R_{w_4} = 0.03\Omega, R_{w_5} = 0.08\Omega, R_{w_6} = 0.07\Omega$.

As the first step of Algorithm 2, the currents of the feeders are computed using Equations (3.2)-(3.3): $I_{F_1} = 11.77$ A and $I_{F_2} = 15.23$ A.

After that, the load where the network should be cut in half can be determined in the *second step.* Then in every iteration the first subnetwork is complemented with new nodes and edges and the current of the next wire is computed:

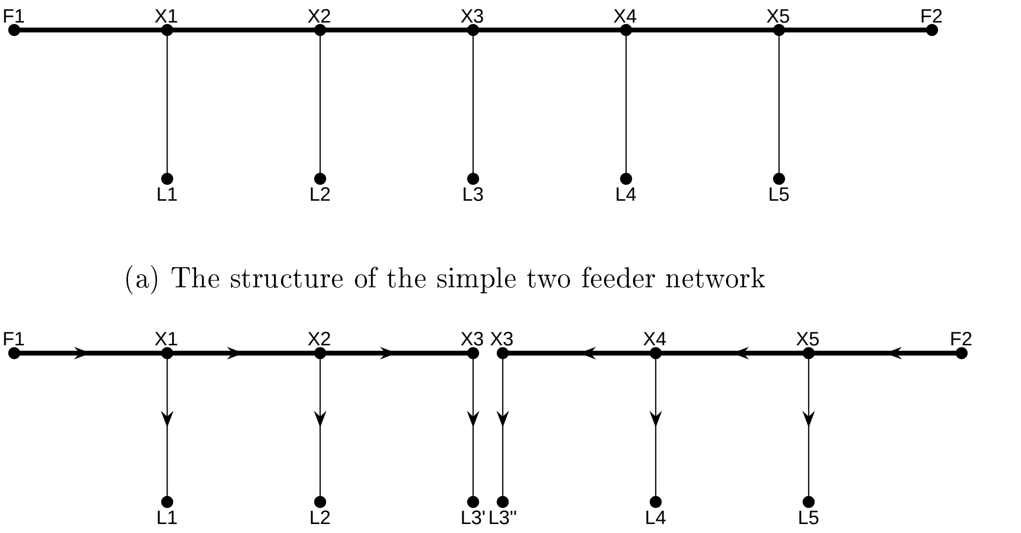
1. initialization: $V_1 = \{F_1\},\ A_1 = \emptyset,\ I_{w_1} = 11.77A,\ i = 1,\ X_0 = F_1$
2. $V_1 = \{F_1, X_1, L_1\},\ A_1 = \{(F_1, X_1), (X_1, L_1)\},$
   $I_{w_2} = I_{w_1} - I_1 = 8.77A,\ i = 2$
3. $V_1 = \{F_1, X_1, L_1, X_2, L_2\},\ A_1 = \{(F_1, X_1), (X_1, L_1), (X_1, X_2), (X_2, L_2)\},$
   $I_{w_3} = I_{w_2} - I_2 = 1.77A,\ i = 3$
4. $V_1 = \{F_1, X_1, L_1, X_2, L_2, X_3, L_3\},$
   $A_1 = \{(F_1, X_1), (X_1, L_1), (X_1, X_2), (X_2, L_2), (X_2, X_3), (X_3, L_3)\},$
   $I_{w_4} = I_{w_3} - I_3 = -3.23A\ i = 4$

It can be seen that I_{w_4} is less than 0, which means the network should be cut in half at the third load. This load is fed by both feeders therefore L_3 is substituted by two new nodes L_3' and L_3''. The current of L_3' is $I_3' = I_3 + I_{w_4} = 1.77A$ and the current of L_3'' is $I_3'' = -I_{w_4} = 3.23A$. The nodes and the connecting arcs are added to the corresponding subnetworks. The result of the decomposition are two subnetworks:

- $S_1 = (V_1, A_1),$
  $V_1 = \{F_1, X_1, L_1, X_2, L_2, X_3, L_3'\},$
  $A_1 = \{(F_1, X_1), (X_1, L_1), (X_1, X_2), (X_2, L_2), (X_2, X_3), (X_3, L_3')\}$
- $S_2 = (V_2, A_2),$
  $V_2 = \{X_3, L_3'', X_4, L_4, X_5, L_5, F_2\},$
  $A_2 = \{(F_2, X_5), (X_5, L_5), (X_5, X_4), (X_4, X_3), (X_3, L_3'')\}$

The two resulting subnetworks are displayed in Figure 3.6b and 3.6c.

Note that *the point of the cut depends on the network parameters* (load currents and wire resistances).



(a) The structure of the simple two feeder network

(b) The first subnetwork of the example

(c) The second subnetwork of the example

Figure 3.6: The simple two feeder network and its decomposition

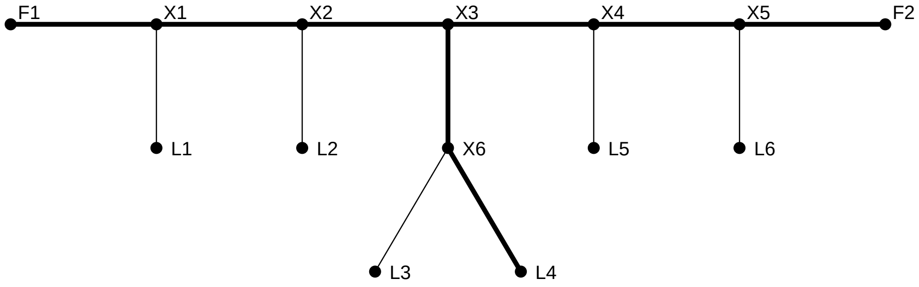


Figure 3.7: The two feeder radial network example.

Two feeder network with radial layout The second example is a two feeder network with radial layout that can be seen in Figure 3.7. The difference from the previous example is that there is a branch in the transmission line at node $X6$. Two loads (nodes $L3$ and $L4$) are connected to this branch with currents of 2 A and 3 A.

The decomposition in this case means the reduction of the network to a basic two feeder layout. To do this at first the loads $L3$ and $L4$ on a branch should be merged into one load. The current of the merged load is the sum of the loads, which is $2 + 3 = 5$ A in this case. So the loads $L3$ and $L4$ are substituted with a merged load at node $X6$. The converted network has the same structure as the two feeder network in the previous example in Figure 3.6a. After that the decomposition into two one feeder networks can be done in the same way as in the previous example.

3.3 Diagnosis of electrical networks

In this section the method of detection and isolation of the illegal users in the network is presented. The fault diagnosis of a system is usually composed of two tasks [8]. In the *detection* task the occurrence of a fault is investigated. If the presence of a faulty behaviour is detected the type of the occurred fault is determined in the *fault isolation* task. In the context of diagnosis of electrical networks with respect to non-technical losses, fault detection and fault isolation has the following meaning:

- Fault detection means that the presence of non-technical losses is noticed.
- Fault isolation aims to determine the location and the magnitude of the non-technical losses. If there are more than one non-technical losses then each location and magnitude should be determined, if possible.

The fault isolation is usually more difficult than the fault detection. The presence of multiple faults adds additional difficulties, as one wants to clearly isolate them. The novelty of the proposed diagnostic method lies in the fact that structural decomposition of the network is used to locate non-technical losses. This enables to isolate multiple NTLs in most of the cases.

The proposed fault detection and isolation method is presented as follows. In Section 3.3.1 the principle of fault detection and fault isolation method is presented. Because of its special theoretical and practical importance, the multiple fault case is discussed in a separate section together with the effect of measurement errors.

The fault diagnosis method is slightly different in case of one feeder and two feeder networks therefore they are presented separately in Section 3.3.2 and Section 3.3.3, respectively. The diagnosis method is illustrated on simple case studies in both cases.

3.3.1 The principle of the method

It is important that it is assumed that the structure and the model of the network is known, and the network is decomposed into simple one feeder structures as introduced in Section 3.2.2. This implies that a *preliminary off-line step of the diagnosis, a powerful electrical decomposition method is assumed to be performed, which breaks down the overall network to subsystems with one feeder layout.*

Then the proposed fault diagnosis method is based on comparing the measured and the nominal values of the network at a given time. The nominal values of the variables at the current time are either known or can be computed using Kirchoff's laws and Ohm's law. The equations used to compute the nominal values are collected here:

- Current of the feeder (one feeder network):

$$I_F = \sum_{i=1}^{N_L} I_i \tag{3.4}$$

- Current of the feeders (two feeder network, from Section 3.2.2):

$$I_{F_1} + I_{F_2} = \sum_{i=1}^{N_L} I_i$$

$$I_{F_2} \sum_{j=1}^{N_W} R_{w_j} = \sum_{i=1}^{N_L} (I_i \sum_{j=1}^{i} R_{w_j})$$

- Current of the wire:

$$I_{w_i} = I_{w_{i-1}} - I_{i-1} \tag{3.5}$$

- Voltage of the node:

$$U_i = U_{i-1} - I_i R_i, \tag{3.6}$$

  where i is the current node, and $i-1$ is its predecessor, I_i and R_i is the current and the resistance of the wire between nodes $i-1$ and i.

The measured values come from the smart meters installed at each load of the network. The proposed fault detection and isolation methods are valid at the current time of the measurement.

The inputs of the diagnosis are:

- Nominal values: currents of the loads and the feeder(s), voltage of the feeder, computed voltages of the loads with a given uncertainty.

- Measured values: currents and voltages of the loads and the feeder(s) with measurement error.

Fault detection

The first part of the diagnosis is the fault detection. The fault detection method works on the whole (not decomposed) network. The measured currents of the loads and the transformer are used to detect the non-technical losses in the network. The measured currents of the loads are denoted by $\tilde{I}_i$, $i = 1 \dots N_L$ and the measured current of the feeder by $\tilde{I}_F$. The error of the current measurement is given in percentages and denoted by ε_I. Then the criterion of detection is that the difference between the sum of the measured currents of the loads and the measured current of the feeder is greater than the maximum measurement error.

$$\tilde{I}_F - \sum_{i=1}^{N_L} \tilde{I}_i > \varepsilon_I(\tilde{I}_F + \sum_{i=1}^{N_L} \tilde{I}_i) \tag{3.7}$$

It was assumed in Section 3.1.2 that the fraud meter shows less current than the real one, while the current of the feeder is equal to the sum of the real currents of the loads. Therefore the current of the feeder should be greater than the currents of the loads in that case if a non-technical loss is present. The maximum measurement error is computed as the sum of the measurement errors of each meter.

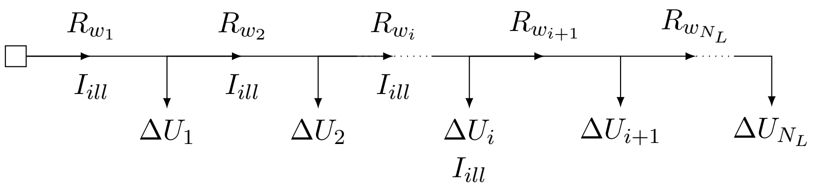


Figure 3.8: One illegal load in the network

Fault isolation: Voltage difference method

The more difficult part of the diagnosis is the fault isolation. The proposed fault isolation method works on the decomposed subnetworks. The proposed method is based on analyzing the difference between the measured and the nominal voltages of successive loads in the decomposed subnetworks.

The principle of the fault isolation method is that the increased current of the illegal loads increases the voltage drop of all loads in the network with respect to the nominal voltages. A similar approach was presented in [77], where the voltage drop of successive nodes was analyzed by statistical methods with respect to previous measurement data. The novelty of my proposed fault isolation method is that the estimated magnitude of the illegal current can be computed knowing the voltage differences and the resistances of the wires. Instead of analyzing the voltage drop, the change in the computed illegal current is taken into account.

Let the measured and the nominal voltages of the loads be denoted by $\tilde{U}_i$, $i = 1 \dots N_L$ and U_i, $i = 1 \dots N_L$. Then the difference between the measured and the nominal voltages at each load can be computed as

$$\Delta U_i = \tilde{U}_i - U_i,\ i = 1 \dots N_L.$$

If there is any non-technical loss in the network the voltage difference has negative value at each load.

The method is introduced with the help of a general example, which can be seen in Figure 3.8.

Let us assume that there are N_L loads in the same transmission line, and let the ith user be an illegal user. Let the extra current of the illegal user be denoted by I_{ill}, so the real current of the illegal user is $I_i + I_{ill}$. The voltage differences at the loads are caused by the I_{ill} current. The voltage differences before the illegal load can be expressed using the illegal current:

$$\begin{aligned}
\Delta U_1 &= -I_{ill} R_{w_1} \\
\Delta U_2 &= -I_{ill}(R_{w_1} + R_{w_2}) \\
&\vdots \\
\Delta U_i &= -I_{ill}(R_{w_1} + \ldots + R_{w_i})
\end{aligned}$$

The voltage differences of the successive loads after the illegal load are equal, because $I_{ill} = 0$ on that section of the transmission line.

$$\Delta U_i = \Delta U_{i+1} = \ldots = \Delta U_{N_L} \tag{3.8}$$

The voltage differences can be expressed using the voltage difference of the previous load:

$$\begin{aligned}
\Delta U_2 &= \Delta U_1 - I_{ill} R_{w_2} \\
&\vdots \\
\Delta U_i &= \Delta U_{i-1} - I_{ill} R_{w_i}
\end{aligned}$$

It can be seen that the magnitude of the illegal current can be computed at every load:

$$\begin{aligned}
I_{ill} &= -\frac{\Delta U_1}{R_{w_1}} \\
I_{ill} &= \frac{\Delta U_1 - \Delta U_2}{R_{w_2}} \\
&\vdots \\
I_{ill} &= \frac{\Delta U_{i-1} - \Delta U_i}{R_{w_i}}
\end{aligned}$$

The voltage differences of the successive loads after the ith load are equal, therefore the computed illegal current is 0 at these loads. It can be seen that the computed illegal currents at the loads are equal before the illegal load, and 0 after it. From the above equations the localization of the illegal load looks simple: the load after that the illegal current is 0 need to be found.

Measurement errors, multiple faults

Measurement errors In practice the effect of measurement errors should be taken into account during the diagnosis. The current measurement error has been already taken into account during the fault detection (see Section 3.3.1). However there are other measurement errors and parameter uncertainties that affects the diagnosis. First of all, the nominal voltages are computed by Equation (3.6) have a lower and an upper bound, which comes from the uncertainty of the resistances.

$$\begin{aligned}
U_{min_i} &= U_{i-1} - I_i(1 + \varepsilon_R) R_{w_i}, \\
U_{max_i} &= U_{i-1} - I_i(1 - \varepsilon_R) R_{w_i},
\end{aligned}$$

where ε_R is the uncertainty of the resistances (in fractions). During the diagnosis it should be checked first, whether the measured voltages are in the range of (U_{min_i}, U_{max_i}). If the measured voltage is between the lower and upper limits, it cannot be decided that the voltage deviation is the result of illegal consumption or the parameter uncertainty.

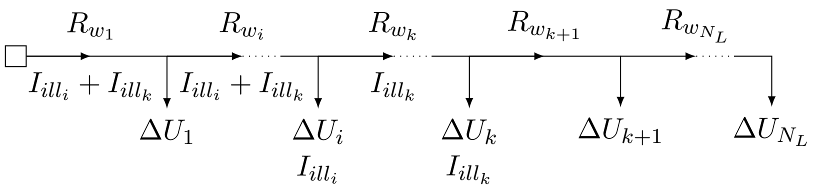


Figure 3.9: Two illegal loads in the network

The other effect of the measurement errors appears in the fault isolation phase. Because of the measurement errors, the computed illegal currents are not the exact values only approximations of the real illegal currents. Therefore a reasonable threshold should be applied, when comparing the computed illegal currents. An evident candidate for the error threshold could be the computational error of the illegal current. It can be determined based on the error of the voltage differences, because the illegal current is computed from these values. The difference between the voltage differences can be written as

$$\Delta U_i - \Delta U_{i+1} = \tilde{U}_i - U_1 - \tilde{U}_{i+1} + U_{i+1}$$

The voltage measurement errors on the right side of the previous formula are

$$\varepsilon_U \tilde{U}_i - 0 - \varepsilon_U \tilde{U}_{i+1} + 0,$$

therefore the error threshold can be chosen to $\varepsilon_U(\tilde{U}_i - \tilde{U}_{i+1})$.

The fault isolation method in that case need to be modified in such a way, that the difference between the successive computed illegal currents should be larger than the error threshold.

Multiple faults If there is only one illegal load in the transmission line, then the location of the illegal load could be determined only by analyzing the voltage differences. The illegal load is that load, from that the voltage differences are equal. But if there are more than one illegal load, only the last one could be localized with that method. Iterating this step, the other illegal loads in the branch can be discovered. The disadvantage of this method is that the network should be simulated after each identified illegal load. Therefore I propose an alternative solution to localize multiple illegal loads.

The fault isolation method introduced in Section 3.3.1 can be used to localize more than one illegal loads in one step in a transmission line. The case of *multiple faults* is introduced through the example of two illegal loads. The example network with two illegal loads can be seen in Figure 3.9.

Let us assume that the ith and the kth are illegal loads. The illegal current of them is denoted by I_{ill_i} and I_{ill_k}, respectively. The voltage differences of the loads can be written as:

$$\begin{aligned}
\Delta U_1 &= -(I_{ill_i} + I_{ill_k})R_{w_1} \\
&\vdots \\
\Delta U_i &= -(I_{ill_i} + I_{ill_k})(R_{w_1} + \ldots + R_{w_i}) = \Delta U_{i-1} - (I_{ill_i} + I_{ill_k})R_{w_i} \\
\Delta U_{i+1} &= -(I_{ill_i} + I_{ill_k})(R_{w_1} + \ldots + R_{w_i}) - I_{ill_k}R_{w_{i+1}} = \Delta U_i - I_{ill_k}R_{w_{i+1}} \\
&\vdots \\
\Delta U_k &= -(I_{ill_i} + I_{ill_k})(R_{w_1} + \ldots + R_{w_i}) - I_{ill_k}(R_{w_{i+1}} + \ldots + R_{w_k}) \\
&= \Delta U_{k-1} - I_{ill_k}R_{w_k} \\
\Delta U_k &= \Delta U_{k+1} = \ldots = \Delta U_{N_L}
\end{aligned}$$

The computed illegal currents at the loads are the following:

$$\begin{aligned}
I_{ill_i} + I_{ill_k} &= -\frac{\Delta U_1}{R_{w_1}} \\
I_{ill_i} + I_{ill_k} &= \frac{\Delta U_1 - \Delta U_2}{R_{w_2}} \\
&\vdots \\
I_{ill_i} + I_{ill_k} &= \frac{\Delta U_{i-1} - \Delta U_i}{R_{w_i}} \\
I_{ill_k} &= \frac{\Delta U_i - \Delta U_{i+1}}{R_{w_{i+1}}} \\
&\vdots \\
I_{ill_k} &= \frac{\Delta U_{k-1} - \Delta U_k}{R_{w_k}} \\
0 &= \frac{\Delta U_k - \Delta U_{k+1}}{R_{w_{k+1}}} = \ldots = \frac{\Delta U_{N_L-1} - \Delta U_{N_L}}{R_{w_{N_L}}}
\end{aligned}$$

To localize the illegal loads the place where the computed illegal current changes its value need to be found. In the example above, there are two such places in the sequence of the illegal currents. The first one is at the ith load: up to this load the illegal current is $I_{ill_i} + I_{ill_k}$ and I_{ill_k} beyond that. The second one is at the kth load: up to that the illegal current is I_{ill_k} and 0 beyond that. The magnitude of the illegal currents can be determined knowing I_{ill_k} and $I_{ill_i} + I_{ill_k}$.

The method of two illegal users can be generalized for more illegal users too. In this case the computation algorithm of the illegal loads consists of the following steps.

1. Compute the difference between the measured and nominal voltages of the loads.

2. Compute the illegal current at each load.
3. Find the places where the magnitude of the illegal current changes.
4. The magnitude of the illegal currents can be computed in reversed order.

3.3.2 Fault detection and isolation for the one feeder radial layout

In this case the fault isolation part of the diagnosis is based on the voltage difference principle introduced in Section 3.3.1. The structural decomposition of the network is also utilized during the diagnosis. Different cases of the diagnosis (for multiple and single load subnetworks) are distinguished according to the number and locations of the illegal loads. The notation of single and multiple load subnetworks introduced in Section 3.2.2 are used here.

The fault detection method is the same as it is described in Section 3.3.1.

Fault isolation in multiple load subnetworks

The first case is when there are one or more illegal loads and they are located in multiple load subnetworks. Then the fault isolation method can be applied separately to each multiple load subnetwork. The advantage of the decomposition is that the fault isolation method can be applied to the subnetworks in parallel and the complex problem can be traced back to a simpler one.

There is only one minor modification in the computation of the voltage differences and the illegal currents in this case. After the decomposition there is no feeder at the start of the subnetwork but the junction node of the branch acts as a pseudo-feeder. Therefore, the voltage of the feeder is substituted by the voltage of the junction node at the start of the branch. Let it be denoted by U_B. If the illegal load is the first one, then the voltage difference of the first load can be computed by:

$$\Delta U_1 = \Delta U_B - I_{ill} R_{w_1}$$

The problem is that the measurement points are at the loads. Therefore the voltage of the junction node (U_B) is not measured therefore ΔU_B cannot be computed. Without knowing the value of ΔU_B the illegal current at the first load cannot be computed. *Therefore it is impossible to localize the illegal load with this method if it is the first load of the subnetwork.* In this case the method presented in the next section can be applied to that load.

Fault isolation in single load subnetworks

It might happen that the illegal load is in a single load subnetwork. The problem is that the single load subnetworks contain only one load therefore the voltage difference cannot be compared to another load in the same subnetwork. In this case the decomposition based method cannot be applied. Instead, the original network is used in the diagnosis.

In this case the voltage difference of the single load is compared to the nearest load in both directions in the network. (It is assumed that these loads are not illegal loads). The steps of the fault isolation method in this case are the following:

1. The single load is denoted by L_s and the voltage difference at this load is ΔU_{L_s}.
2. Go back to the previous junction from the single load. This node is the predecessor of the single load in the directed graph.
3. Find the nearest load from that node backwards in the graph. This load is denoted by L_b and the voltage difference at this load is ΔU_{L_b}.
4. Find the nearest load from that load forward in the graph. This load is denoted by L_f and the voltage difference at this load is ΔU_{L_f}.
5. If $\Delta U_{L_b} > \Delta U_{L_s} < \Delta U_{L_f}$ then L_s is considered as an illegal load.

The exact value of the illegal current can be computed in this case if there are no additional illegal loads in the network. The magnitude of the illegal current is $\frac{\Delta U_{L_b} - \Delta U_{L_s}}{R_{w_{bs}}}$, where $R_{w_{bs}}$ is the resistance of the wire between the start of the branch of the L_b load and the single load. If there are more illegal loads, then the computed current may not be accurate because it may be affected by the currents of the other illegal loads.

Diagnostic algorithm for one feeder radial networks

The diagnostic algorithm of a one feeder radial network combines the detection of the illegal loads and the localization of them whether they are in multiple or single load subnetworks. There are two main parts of the diagnosis method.

- First the illegal loads in *multiple load subnetworks* are localized using the method described in Section 3.3.2. Then the currents of the found illegal loads are updated with the currents of the illegal loads, and the network is simulated with the updated currents. In these steps all illegal loads in the multiple load subnetworks are discovered.
- If the criterion of the detection is still satisfied after the simulation then there are remaining illegal loads in the network. They should be in the *single load subnetworks*. These illegal loads are localized in the second part of the algorithm using the method described in Section 3.3.2.

The steps of the complete diagnostic method are the following.

1. Decompose the radial feeder network into single and multiple load subnetworks.
2. Compute the nominal voltages and the minimal/maximal bounds of the voltages using the measured currents and the resistances.

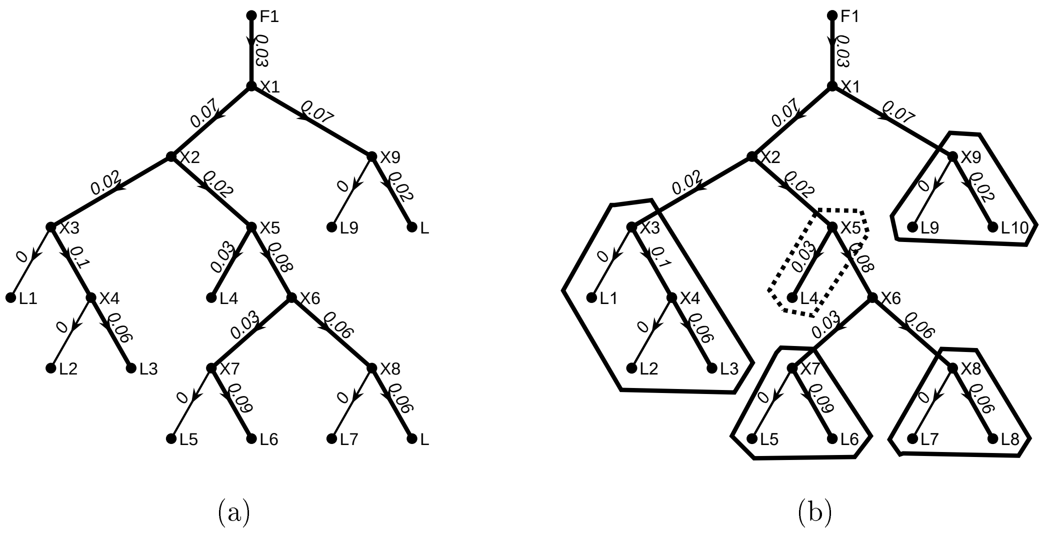


Figure 3.10: A simple one feeder radial network and its decomposition.

3. Detect the illegal loads using Equation (3.7). If the inequality is true then continue with analyzing the voltage differences.

4. Compute the difference between the measured and the nominal voltages.

5. If the difference is outside of the minimum/maximum bounds then continue with the localization of the illegal loads.

6. Find illegal loads in multiple load subnetworks.

7. Update the currents of the found illegal users with the computed illegal current.

8. Compute the nominal voltages and the minimum/maximum bounds using the updated currents. Attach the computed values to the subnetworks. The network does not need to be decomposed again.

9. Repeat Steps 3-5.

10. Find the illegal loads in single load subnetworks.

A simple one feeder radial network example

The diagnostic method is illustrated on a simple example. The network can be seen in Figure 3.10a. It is the same network as in Section 3.2.2. The decomposition of the network can be seen in Figure 3.10b. The resistances of the wires can be seen on the edges of the graph. The uncertainty of the resistances is 2%, the current and voltage measurement error is 0.2%. The measured currents and voltages of the feeder and the loads are in Table 3.1 in the rows $\tilde{I}$ and $\tilde{U}$.

Table 3.1: Current and voltage values of the example.

	$F1$	$L1$	$L2$	$L3$	$L4$	$L5$	$L6$	$L7$	$L8$	$L9$	$L10$
$\tilde{I}$[A]	63	5	4	8	8	2	5	5	7	8	8
$\tilde{U}$[V]	250	244.44	243.04	242.56	243.99	242.53	242.08	242.02	241.60	246.99	246.83
U[V]	250	244.78	243.58	243.10	244.34	242.85	242.40	242.34	241.92	247.08	246.92
ΔU[V]	0	-0.34	-0.54	-0.54	-0.35	-0.32	-0.32	-0.32	-0.32	-0.09	-0.09

The steps of the diagnostic algorithm, the used and computed data and the results of the steps in case of the example can be seen in Table 3.2. It can be seen, that after the first part of the diagnosis (Steps 1-6) one illegal load ($L2$) is found in a multiple load subnetwork with illegal current of 2A. After that the network is updated and simulated with this current (Steps 7-8). The computed voltages after Step 8 can be seen in Table 3.3. In the second part of the diagnosis (Steps 9-10) an other illegal load ($L4$) is found in a single load subnetwork with illegal current of 1A. In conclusion our diagnostic method successfully detected and localized these loads, and determined the magnitude of the illegal currents too.

A more complex case study can be found in Appendix A where the diagnostic method is demonstrated on a benchmark example. The case study shows the application of the decomposition and diagnostic algorithms in an almost real-world case, with simulated measurement errors.

3.3.3 Fault detection and isolation for the two feeder layout

Diagnostic algorithm for two feeder networks

The principle of the diagnosis can be applied to networks with two feeder layout, too. In this case the network should be first simplified to a basic two feeder layout (see Section 3.2.2). The measured current of a merged branch is the sum of the measured currents in the branch. The measured voltage of a branch is substituted by the measured voltage of the first load in the branch. It is assumed that the resistance of the wire connecting the merged load and the network can be neglected.

To *detect the illegal loads* the condition in Equation 3.7 should be modified to use the currents of both feeders:

$$\tilde{I}_{F_1} + \tilde{I}_{F_2} - \sum_{i=1}^{N_L} \tilde{I}_i > \varepsilon_I (\tilde{I}_{F_1} + \tilde{I}_{F_2} + \sum_{i=1}^{N_L} \tilde{I}_i) \tag{3.9}$$

Then the voltage differences and the illegal currents can be computed starting from one of the feeders.

In case of the two feeder network the illegal current changes its sign at some point. This is the place of the illegal load.

Table 3.2: The diagnosis steps of the example.

<table>
<tr><th>Step</th><th>Action</th><th colspan="10">Data</th><th>Result</th></tr>
<tr><td>(1)</td><td>Decomposition</td><td colspan="10">graph of the network</td><td>subnetworks</td></tr>
<tr><td>(2)</td><td>Simulation</td><td colspan="10">$\tilde{I}_F$, $\tilde{I}_i$, R_{w_i}, $\varepsilon_R = 2\%$</td><td>U,U_{min},U_{max}</td></tr>
<tr><td>(3)</td><td>Detection</td><td colspan="10">$\tilde{I}_F = 63\text{A}, \sum_{i=1}^{10} \tilde{I}_i = 60\text{A}$, $\varepsilon_I = 0.2\%$</td><td>TRUE</td></tr>
<tr><td>(4)</td><td>Compute voltage difference</td><td colspan="10">U,$\tilde{U}$, U_{min}, U_{max}</td><td>ΔU, ΔU_{min}, ΔU_{max}</td></tr>
<tr><td>(5)</td><td>Check voltage difference</td><td colspan="10">ΔU, ΔU_{min}, ΔU_{max}</td><td>TRUE</td></tr>
<tr><td>(6)</td><td>Localization I.</td><td></td><td>$L1$</td><td>$L2$</td><td>$L3$</td><td>$L5$</td><td>$L6$</td><td>$L7$</td><td>$L8$</td><td>$L9$</td><td>$L10$</td><td>NTL=$L2$</td></tr>
<tr><td></td><td>(multiple load subnetworks)</td><td>I_{ill}[A]</td><td>-</td><td>2</td><td>0</td><td>-</td><td>0</td><td>-</td><td>0</td><td>-</td><td>0</td><td>$I_{ill} = 2\text{A}$</td></tr>
<tr><td>(7)</td><td>Update</td><td colspan="10">$\tilde{I}_2 = 4\text{A}$, $I_{ill} = 2\text{A}$</td><td>$\tilde{I}_2 = 6\text{A}$</td></tr>
<tr><td>(8)</td><td>Simulation</td><td colspan="10">$\tilde{I}_F$, $\tilde{I}_i$, R_{w_i}, $\varepsilon_R = 2\%$</td><td>U_i;U_{min},U_{max}</td></tr>
<tr><td>(9-3)</td><td>Detection</td><td colspan="10">$\tilde{I}_F = 63\text{A}$, $\sum_{i=1}^{10} \tilde{I}_i = 62\text{A}$, $\varepsilon_I = 0.2\%$</td><td>TRUE</td></tr>
<tr><td>(9-4)</td><td>Compute voltage difference</td><td colspan="10">U,$\tilde{U}$, U_{min}, U_{max}</td><td>ΔU, ΔU_{min}, ΔU_{max}</td></tr>
<tr><td>(9-5)</td><td>Check voltage difference</td><td colspan="10">ΔU, ΔU_{min}, ΔU_{max}</td><td>TRUE</td></tr>
<tr><td>(10)</td><td>Localization II.</td><td></td><td>$L1$</td><td>$L4$</td><td>$L5$</td><td colspan="6">single load: $L4$, $R = 0.05\Omega$</td><td>NTL=$L4$</td></tr>
<tr><td></td><td>(single load subnetworks)</td><td>ΔU[V]</td><td>-0.1</td><td>-0.15</td><td>-0.12</td><td colspan="6">neighbours:$L1, L5$</td><td>$I_{ill} = 1\text{A}$</td></tr>
</table>

Table 3.3: Current and voltage values after the simulation.

	$F1$	$L1$	$L2$	$L3$	$L4$	$L5$	$L6$	$L7$	$L8$	$L9$	$L10$
$\tilde{I}$[A]	63	5	6	8	8	2	5	5	7	8	8
$\tilde{U}$[V]	250	244.44	243.04	242.56	243.99	242.53	242.08	242.02	241.60	246.99	246.83
U[V]	250	244.54	243.14	242.66	244.14	242.65	242.20	242.14	241.72	247.02	246.86
ΔU[V]	0	-0.1	-0.1	-0.1	-0.15	-0.12	-0.12	-0.12	-0.12	-0.03	-0.03

If there are more than one illegal loads then this is the place of the largest illegal load. The other illegal loads can be found where the illegal current changes its value. If the result of the fault isolation step is a merged branch then the illegal load is in that branch. The exact location of it can be determined by applying the diagnostic algorithm of the radial network to the merged branch.

In summary, the diagnostic method consists of the following steps.

1. Convert the two feeder network into a basic two feeder layout.
2. Simulate the converted network with the measured currents and compute the nominal voltages.
3. Use Equation (3.9) to detect the presence of illegal loads.
4. Compute the difference between the measured and the nominal voltages.
5. Compute the sequence of the illegal currents $I_{ill_i} = \frac{\Delta U_i - \Delta U_{i+1}}{R_{i+1}}$, $i = 1, \ldots, N_L - 1$.
6. Find the places where the illegal current changes its value, e.g. $I_{ill_i} \neq I_{ill_{i+1}}$. The illegal load is the $i+1$th load.
7. Compute the value of the illegal currents of the found loads: $I_{ill_{i+1}} = I_{ill_i} - I_{ill_{i+1}}$.
8. If the illegal load is a merged branch, then apply the diagnostic method of the one feeder radial network to this branch to find the exact location of the illegal loads.

Simple example

The diagnostic method is illustrated on the simple example used at the decomposition in Section 3.2.2. The network with the edge weights can be seen in Figure 3.11. This is not a basic two feeder layout therefore a conversion is needed first to obtain a simple two feeder network layout. The converted network has the same structure as shown in Figure 3.6.

The measured currents and voltages of the converted network can be seen in the first two rows of Table 3.4. The steps of the diagnosis of the example can be seen in Table 3.5. It can be seen that the computed illegal currents change value at two loads: $L2$ and $X6$ (Step 5-6). Therefore they are both illegal

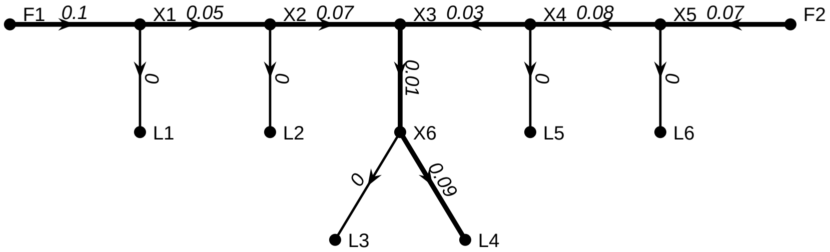


Figure 3.11: The two feeder network of the example.

loads with illegal currents 1A and 2A respectively. It can be seen that $X6$ is a substitution of a branch, therefore the illegal load is in that branch. The measured and nominal voltages of this subnetwork can be found in Table 3.6. The illegal load in this subnetwork is localized in Step 8. In conclusion two illegal loads are successfully detected and localized in the two feeder network.

Table 3.4: Measured and nominal values of the two feeder example.

	$F1$	$L1$	$L2$	$X6$	$L5$	$L6$	$F2$
$\tilde{I}$[A]	13.3	3.0	7.0	5.0	4.0	8.0	16.7
$\tilde{U}$[V]	250.0	248.67	248.16	247.99	248.14	248.83	250.0
U[V]	250.0	248.82	248.38	248.26	248.36	248.93	250.0
ΔU[V]	0	-0.15	-0.22	-0.27	-0.22	-0.1	0

Table 3.5: Diagnosis steps of the two feeder example.

Step	Action	Data						Result
(1)	Network conversion	graph of the network						basic two feeder
(2)	Simulation	$\tilde{I}_{F1}$, $\tilde{I}_{F2}$, $\tilde{I}_i$,R_{w_i}, $\varepsilon_R = 2\%$						U_i
(3)	Detection	$\tilde{I}_{F_1} = 13.3$A, $\tilde{I}_{F_2} = 16.7$A, $\sum_{i=1}^{6} \tilde{I}_i = 27$A, $\varepsilon_I = 0.2\%$						TRUE
(4)	Compute voltage difference	$\tilde{U}_i$, U_i						ΔU_i
(5)	Compute illegal currents		$L1$	$L2$	$X6$	$L5$	$L6$	sequence of
		I_{ill}[A]	-	1.53	0.53	-1.47	-1.47	illegal currents
(6)	Localization I.			x	x			NTL=$L2, X6$
(7)	Compute illegal currents of $L2$ and $X6$	1.53A, 0.53A 0.53A, -1.53A						$I_{ill,L2} = 1$A $I_{ill,X6} = 2$A
(8)	Localization II.		$L3$	$L4$				NTL=$L4$
		I_{ill}[A]	-	2				$I_{ill,L4} = 2$A

Table 3.6: Measured and nominal values of the subnetwork

	$L3$	$L4$
$\tilde{U}$[V]	247.99	247.63
U[V]	248.26	248.08
ΔU[V]	-0.27	-0.45

3.4 Discussion and future work

As it was mentioned earlier, the proposed diagnostic method is similar to the power theft localization method described in [77], which is called *normalized voltage double difference technique (NVDD)*. Both methods are based on the analysis of the voltage differences. However there are main differences between the two methods:

- The NVDD method uses phasor values (i.e. complex voltage, current and impedance values). However the phase angel between successive nodes is usually really small, therefore the voltage drop can be approximated by the real part of the impedance drop. In my method I use real valued variables.
- The NVDD method uses measurements from reference days as nominal voltage values, and detects the outliers compared to these data. In my method a static model of the network is used to obtain the nominal values of the network instead of historical data.
- In the NVDD method the voltage differences are normalized by the variations of the voltage differences. In my method the illegal currents of successive loads are computed and analyzed to determine the location of the illegal load.

The advantages of my presented method are:

- The disadvantage of the NVDD method is that it requires sufficient reference days, which may be hard to obtain (e.g. consumers with PV plants may cause a variation in the load profiles). The advantage of using a static network model is that it always gives the current state of the network. Therefore variations in the load profiles has no effect on the efficiency of the method. Repeating the simulation and the diagnosis at consecutive time instances, the dynamic behaviour of the loads can be inferred.
- In contrast to the NVDD method, the illegal loads that appear after the beginning of the operation of the diagnostic system can be localized with my method.
- Reference day measurements are not needed, because the nominal values are computed by the network model.

- The presence of multiple illegal loads and networks with two feeders are not discussed in [77]. My proposed method is able to localize multiple illegal loads and can be applied to two feeder networks, too.
- Multiple loads can be localized in one step.

The limitation of my method are:

- If the illegal load is in the single load subnetwork, and its neighbours are also illegal loads, then it cannot be localized.
- Both methods depend on the measurement error. With the increase of the measurement errors, one obtains more false illegal loads, but the real illegal loads can still be detected and localized.
- The diagnostic accuracy is also influenced by the error threshold limit for the computed illegal load current and the computed voltage differences. With too low thresholds one obtains spurious illegal loads, while the illegal loads cannot be localized using too loose limits.

Future work In the future work the effect of measurement error and parameter uncertainties on the diagnostic accuracy should be analyzed. Moreover in the presented method it was assumed, that the voltage measurements are correct, and the current measurements can be manipulated. However there exist such power theft methods, where the current measurements are not manipulated. An interesting question is that how the basic principles of the method could be applied to that case (e.g. using Kirchoff's voltage law for the detection and computing the nominal currents for the localization).

Moreover, the effect of domestic power plants can also be investigated. Nowadays PV plants become more common in low voltage networks, that can be considered as either loads with negative consumption or generators (i.e. feeders). These power plants may change the network structure (more than two feeders are present) and the diagnostic method should be adapted to this case. The application to three-phase network can also be a possible further research direction.

3.5 Summary

In this chapter the detection and localization of non-technical losses in electrical networks were introduced. The electrical network is composed of feeders, loads and transmission lines that is represented by a directed graph. A simple static linear model of the network was used to compute the network variables, knowing the measured currents of the loads and feeders, and the resistances of the transmission lines (with parameter uncertainties). The measured values come from the smart meters that are located at every load and provide consumption data with measurement errors. The illegal loads, which we want to detect and localize, have fraud meters that show less than the real consumption.

The aim of the work was to create a model-based diagnostic method to detect and localize one ore more illegal loads in the electrical network. The illegal loads are located in different parts of the network. The diagnostic method should estimate the magnitude of the illegal consumption too. The measurement errors and parameter uncertainties were also taken into account.

To avoid the development of a complex global diagnostic algorithm, the decomposition of the network was utilized in the proposed method. With the help of the decomposition the illegal loads can be isolated locally in the decomposed subnetworks and the diagnosis can be performed on them in parallel. In Section 3.2 different kinds of network structures were introduced and two algorithms were developed that successfully decompose the networks with radial or two feeder layouts into smaller subnetworks with basic one feeder layouts.

The decomposition of the radial network can be done with Algorithm 1 that results in single or multiple load subnetworks that contain one or more loads respectively. The decomposition of the two feeder network described in Algorithm 2 is different from that point of view, that the decomposition process is done in parallel with the computation of the network variables.

The diagnostic method is based on analyzing the deviations between the measured and the nominal values of the network. Two kinds of diagnostic methods were introduced that apply to the radial or the two feeder networks. Both methods use the measured and nominal currents for detection and the voltages for fault isolation.

In the diagnostic algorithm for a network with one feeder radial layout the illegal loads were localized in two stages: at first in the multiple load subnetworks then in the single load subnetworks. In the diagnostic method for two feeder networks the localization was also done in two stages: at first in the (converted) basic two feeder network, then in the potential merged subnetworks (in case of a radial two feeder network).

The proposed decomposition and the diagnostic algorithms were illustrated on simple examples. A more complex case study with the IEEE 2015 Low Voltage Test Feeder (that can be found in Appendix A) was also presented to demonstrate the practical application of the diagnostic method.

Chapter 4

Colored Petri net based diagnosis of process systems

In this Chapter a diagnostic method is introduced that uses a qualitative dynamic model of the system and its colored Petri net model. The diagnosis is based on the deviations between normal and faulty operations and the occurrence graph of the colored Petri net model. In case of composite systems structural decomposition is used to reduce the increasing computational effort caused by the growing size of the model.

In many cases the normal or faulty operations of technological processes can be characterized by a series of events having discrete or qualitative valued variables. The occurring deviations can be generated by the comparison of the normal and the actual events. The occurring faults can be detected and identified based on the observed deviations [92], [93].

In order to formally describe the events for diagnosis, the methods and tools for discrete event dynamic systems are used [3], [8]. Technological systems can often be represented as discrete event systems (DES). The solution (i.e. the state space) of DES is usually generated by discrete event simulation. The problem is that the state space of DES models can be extremely large even if the system is relatively simple. Since the diagnosis of DES is usually based on the exploration of the state space, it can be computationally hard task due to the rapid increase of the state space. The decomposition of the system and applying distributed diagnosis can help to solve this problem [94]–[96].

Different kinds of Petri nets are popular tools for representing discrete event systems. The structural and mathematical representation of Petri nets both can be used for diagnostic purposes. Various techniques can be used for diagnosis with Petri nets, for example the analysis of the occurrence graph, marking estimation, linear algebra, integer linear programming, diagnoser nets and reverse nets. The most frequently used methods are based on the idea of unobservable transitions and use labeled Petri net models. Besides the observability of transitions, the set of places may have observable and unobservable subsets, too. In [97] sufficient conditions of diagnosability are given and an on-line fault detection algorithm is developed based on ILP and checking the fault diagnosability conditions. To take into account the firing times of the transitions, the ILP based diagnoser algorithm was extended with timing constraints in [98]. The faults may affect the firing times of transitions, too. In [99] an observer scheme was designed to generate the residuals that can be used for fault detection and isolation. In [100] the unobservable events may correspond to the normal behaviour, too. The diagnosis is performed on a so called basis reachability graph that contains the markings reachable with the observed transitions and the necessary unobservable ones. Colored Petri nets were used to model and diagnose embedded systems in [101]. The inverse CPN model with backward reachability was used to determine the source of the occurred faults in the system.

In this Chapter a novel colored Petri net (CPN) based diagnostic method of technological systems is presented. The novelty of the method is that a general CPN model was constructed that is able to simulate the technological system and simultaneously generate deviations between traces. An on-line diagnostic method is proposed that is based on searching nodes with specific attributes on the occurrence graph of the CPN model. Moreover, structural decomposition of composite systems is proposed, which traces back the diagnosis of composite systems to unit-wise diagnosis.

The structure of this Chapter is the following. At first the basic notions on ordinary and colored Petri nets are given in the next section. Thereafter the colored Petri net model used for diagnosis is introduced in Section 4.2, then the diagnostic methods for single technological units and composite systems are presented in Section 4.3 and Section 4.4. A case study illustrating the proposed diagnostic method can be found in Appendix B.

4.1 Basic notions

A brief introduction of the basic concepts used in this Chapter is given here. At first the concepts of qualitative discrete event models are presented followed by the introduction of the basics of ordinary and colored Petri nets.

4.1.1 Qualitative range sets, events, traces and deviations

In many cases it is not necessary to know the exact value of a variable, it is enough to know whether a measured value is in a range specified in advance. For example, for a continuous sensor S the following basic range values can be defined if the rough resolution is enough:

$$Q_S = \{e^-, 0_S, L_S, N_S, H_S, e^+, error\} \tag{4.1}$$

where 0_S, L_S, N_S, H_S refer to zero, low, normal and high value measured by the sensor S, respectively, while e^- and e^+ may refer to outlier values caused by a bias failure and *error* denotes the error signal provided by the sensor when it is not working. During qualitative simulation it may occur that the range of a variable cannot be decided exactly due to the growing uncertainty caused by the interval arithmetic operations. Therefore in some cases we allow the qualitative variables to be in the union of two neighbouring ranges, too. For example the range value covering L_S and N_S is denoted by LN_S.

The qualitative range value set of binary state actuators is as follows:

$$Q_B = \{op, cl, ?\} \tag{4.2}$$

where *op* and *cl* refers to the opened or closed state while ? denotes the unknown state of the actuator.

Traces The dynamic evolution of a course of a process system with n inputs and m outputs can be characterized by time dependent variables called signals. *Input variables* (var_{in}) are the actions performed by the operators and *output variables* (var_{out}) are the measured values. An *event* occurring at time τ is an ordered list of a time stamp and the values of input and output variables belonging to this time instance:

$$event^{\tau} = (\tau, var_{in_1}(\tau), \ldots, var_{in_n}(\tau), var_{out_1}(\tau), \ldots, var_{out_m}(\tau)).$$

In this work, the time stamp denotes the order of the events instead of the real occurrence time of the event. The real occurrence times can be unevenly spaced values. The event list or operational procedure contains the possible events during a course of a process system. The set of consecutive events is called *trace.*

$$trace = (event^1, event^2, \ldots, event^{\tau})$$

The traces can be categorized into three main groups:

- the *nominal trace* describes the normal (fault-free) operation of the system;
- the *characteristic traces* are specific to each different faulty operational mode of the system;
- the *measured trace* is the actual recorded trace from the process.

Fault diagnosis methods are often based on the comparison of a nominal trace and a measured trace. If there is a *deviation* between these traces then the system works probably in a faulty mode. The deviations can be temporal or quantitative types. The deviations occurring during the process are listed in the form of *keyword(event$^{\tau}$).* Assuming that the events in a trace are ordered by the time stamps, the most important deviation types (and their keywords) are the following:

- never-happened (NH)- if the coherent input-output values of the event do not occur in the characteristic trace at any time stamp;
- chronological deviations: later (LAT) or earlier (EAR) - if the coherent input-output values of the event are in the characteristic trace but with a later or earlier time stamp than in the nominal trace;
- quantitative deviations: greater (GRE) or smaller (SML) - if the value of an output variable is greater or smaller in the characteristic trace than in the nominal trace at a given time instant.

4.1.2 Petri nets

Petri nets are graphical and mathematical modelling tools that are often used for modelling discrete event and discrete time systems. The advantage of Petri nets is that they not only represent the structure of the system, but allows us to simulate and analyze its behaviour too [102]. Petri nets are often used to model such kinds of systems where parallel and concurrent events may occur. Petri nets are represented with directed, weighted, bipartite graphs where two types of nodes are used: places and transitions. Transitions represented by rectangles refer to the events occurring in the system. Places represented by circles refer to precondition and consequences of the events. To follow the changes in the system, tokens (represented by black dots) can be assigned to each place by a *marking function.* A token in a place means that the precondition or consequence represented by the place is "true". The state of the Petri net can be represented by the *marking vector*, that is composed of the markings of the places. When a transition fires the tokens from its input places disappear and the conditions become false. Meanwhile tokens appear on the output places of the transition and the consequences become true. The formal definition of a Petri net (PN) is the following [102].

$$PN = \{P, T, F, W, \underline{\mu}^0\}$$

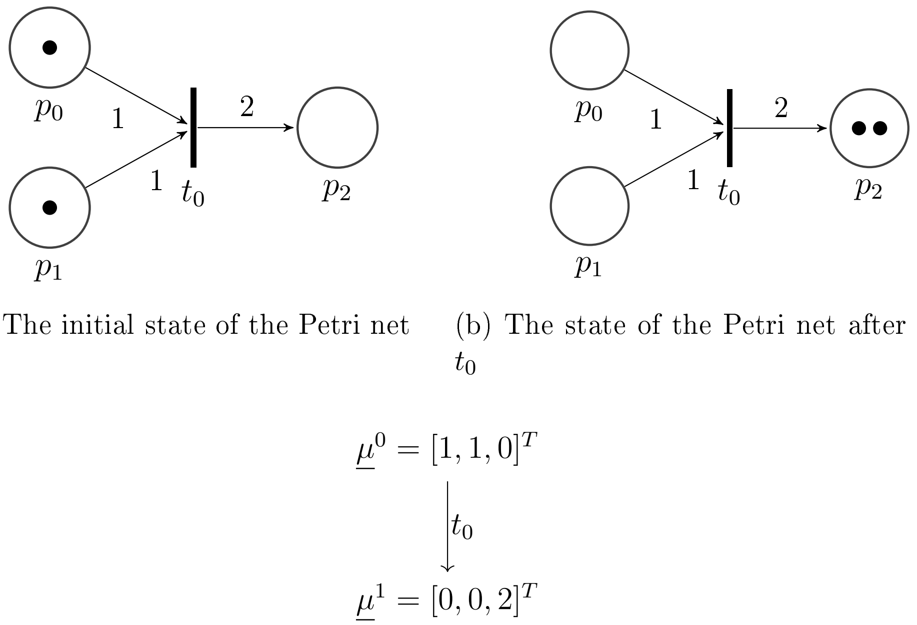
where

- $P = \{p_0, p_1, \ldots, p_m\}$ is the finite set of places;
- $T = \{t_0, t_1, \ldots, t_n\}$ is the finite set of transitions;
- $F \subseteq (P \times T) \cup (T \times P)$ is the set of arcs;
- $W = F \to \mathbb{Z}^+$ is the weight function that assigns a positive integer to the arcs;
- $\underline{\mu}^0 = [\mu_1, \mu_2, \ldots, \mu_m]^T$ is the initial marking vector, where $\mu_i \in \mathbb{N}$ is the number of tokens in the ith place;
- $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$.

Firing of transitions The dynamic operation of a Petri net can be described by the firing sequence of the transitions that move tokens between the places. A transition t_j is enabled and may fire when there are at least $W(p_i, t_j)$ tokens on all of its input places (i.e. all preconditions of the event are true):

$$\mu_i \geq W(p_i, t_j), \forall i \text{ where } p_i \text{ is the input of } t_j.$$

The firing of a transition removes tokens from its input places and put tokens in its output places according to the weight function that leads to a new marking vector. The firing step of t_j that changes the marking vector $\underline{\mu}^i$ to $\underline{\mu}^{i+1}$ is denoted by $\underline{\mu}^i[t_j > \underline{\mu}^{i+1}$. The firing sequence of more transitions is denoted by $\underline{\mu}^0[t_{j0} > \underline{\mu}^1[t_{j1} > \ldots [t_{jk-1} > \underline{\mu}^k$. If there exist such a firing sequence that leads from $\underline{\mu}^0$ to $\underline{\mu}^k$ then we say that $\underline{\mu}^k$ is *reachable* from $\underline{\mu}^0$.



p_0
1
2
p_1
1
t_0
p_2
p_0
1
2
p_1
1
t_0
p_2
$\underline{\mu}^0 = [1,1,0]^T$
t_0
$\underline{\mu}^1 = [0,0,2]^T$

(a) The initial state of the Petri net

(b) The state of the Petri net after firing t_0

(c) The reachability graph of the Petri net

Figure 4.1: A simple Petri net and its reachability graph

Reachability graph The state of the Petri net is characterized by its current marking vector. The possible system states that may occur in the system can be represented on a graph. The *reachability graph* of a Petri net is a directed graph whose nodes are marking vectors and the edges are firings of transitions. There is a directed edge between $\underline{\mu}^k$ and $\underline{\mu}^l$ if $\underline{\mu}^k[t_j > \underline{\mu}^l$. $\underline{\mu}^l$ is called reachable from $\underline{\mu}^k$, if there exists a firing sequence that transforms $\underline{\mu}^k$ to $\underline{\mu}^l$. The reachability graph contains all of the marking vectors (system states) that are reachable from a given initial marking vector.

Simple example A simple example showing the operation of a Petri net can be seen in Figure 4.1. This simple Petri net is composed of three places ($P = \{p_0, p_1, p_2\}$) and one transition ($T = \{t_0\}$). The set of arcs is $F = \{(p_0,t_0),(p_1,t_0),(t_0,p_2)\}$. The value of the weight function is $W(p_0,t_0) = W(p_1,t_0) = 1$ and $W(t_0,p_2) = 2$. The initial marking of the Petri net, which is $\underline{\mu}^0 = [1,1,0]^T$, can be seen in Figure 4.1a. There is one token in p_0 and p_1 therefore t_0 is enabled and can fire. The firing of t_0 removes one tokens from p_0 and p_1 and puts two tokens into p_2. The current state of the Petri net after firing t_0 can be seen in Figure 4.1b. At this state t_0 is not enabled any more because there are no tokens in places p_0 and p_1. The reachability graph of this simple Petri net contains only two nodes, that can be seen in Figure 4.1c.

4.1.3 Colored Petri nets

Colored Petri nets (CPN) are the extensions of the ordinary Petri nets. The advantage of CPNs that complex systems can be represented in a compact form, which leads to the reduced size of the model. In contrast to the ordinary Petri nets, where all tokens are the same, in CPNs the tokens may have different types and attributes, that is described by so called colors. Besides that, functions can be attached to arcs and transitions that execute different operations on tokens. Another option is to fire the transitions with predefined probability that makes the system operation stochastic.

Before giving the formal definition of a CPN some preliminary definitions about multisets and variables are needed.

Multisets A multiset is the extension of an ordinary set in such a way that multiple occurrences of the elements are allowed. It can be defined over a set S as a function that attaches the number of appearances to the elements of S: $m : S \to \mathbb{N}$. The multiset can be represented by a formal sum :$\sum_{s \in S} m(s)`s$, where ` is an explicit operator between the coefficients and the elements. The set of all multisets over S is denoted by S_{MS} and $m(s)$, $s \in S$ is the coefficient of element s.

Different operations can be defined over multisets that are in fact operations on functions:

- addition: $m_1 + m_2 = \sum_{s \in S}(m_1(s) + m_2(s))`s$;
- scalar multiplication: $n \cdot m = \sum_{s \in S}(n \cdot m(s))`s$;
- comparison:$m_1 \neq m_2 = \exists s \in S : m_1(s) \neq m_2(s)$
  $m_1 \leq (\geq, =) m_2 = \forall s \in S : m_1(s) \leq (\geq, =) m_2(s)$
- subtraction: $m_2 - m_1 = \sum_{s \in S}(m_2(s) - m_1(s))`s$, if $m_2 \geq m_1$;

Formal definition As it was mentioned before, different expressions and functions can be assigned to the transitions and arcs of colored Petri nets. The expressions can be defined using variables and different operations (logical, arithmetical). Some basic notations corresponding to variables and expressions are given here:

- TN is a type name;
- $Type(v)$ is the type of the variable v;
- $Type(expr)$ is the type of the expression $expr$;
- $Var(expr)$ is the set of variables in expression $expr$;
- b is a *binding* that is a function mapping each variable $v \in V$ into a value $b(v) \in Type(v)$. Bindings usually denoted by $< v_1 = c_1, \ldots, v_n = c_n >$ where v_i and $c_i, i = 1, \ldots, n$ are the variables and the corresponding values.

- $expr < b >$ is a value, obtained by evaluating the expression *expr* in a binding b

Now the formal definition of a CPN can be given in the following form first introduced by Kurt Jensen [103]. The CPN is a tuple

$$CPN = (\Sigma, P, T, A, N, C, G, E, I)$$

with the following meaning:

- Σ is a finite non-empty set of color sets;
- P is the finite set of places;
- T is the finite set of transitions;
- A is the finite set of arcs, such that $P \cap T = P \cap A = T \cap A = \emptyset$;
- N is the node function, $N : A \to (P \times T) \cup (T \times P)$ that assigns arcs into place-transition or transition-place pairs;
- C is the color function, $C : P \to \Sigma$ that assigns a color set to each place;
- G is the guard function that assigns a boolean expression ($Expr$) to each transition, such that the variables in the guard must belong to a color set in Σ,
  $G : T \to Expr$, such that $Type(G(t)) = bool$ and $Type(Var(G(t))) \subseteq \Sigma$;
- E is the arc expression function that assigns expressions ($Expr$) to each arc, $E : A \to Expr$, such that $\forall a \in A : Type(E(a)) = C(p(a))_{MS}$ and $Type(Var(E(a))) \subseteq \Sigma$, where $p(a)$ is the place of $N(a)$;
- I is the initialization function that defines number and color of initial tokens in the places: $I : P \to Expr$ such that $\forall p \in P : Type(I(p)) = C(p)_{MS}$.

Firing of transitions The enabling and the firing of the transitions in a CPN is more difficult than in ordinary Petri nets. The difference is that the colors of tokens in the places should be taken into account. A transition in a CPN is enabled, if there are tokens on its input places with the proper colors such that binding the variables in the arc expressions to the available token colors, the arc expression can be evaluated. Besides that, the arc expression of the output arc should be evaluated to a multiset that is compatible with the color set of the output place. Moreover the guard function must be evaluated true with this binding of variables. Consequently the enabling of a transition is always interpreted with respect to a specific binding.

The marking M in a CPN is interpreted as the number of colored tokens in the places that is a multiset over the set of all (p, c) pairs, where $p \in P$ and $c \in C(p)$. A marking vector M can be written as $M = [M(p_0), M(p_1), \ldots, M(p_n)]^T$, where $M(p_i)$ denotes the token distribution in place p_i. In contrast to the ordinary Petri nets, a firing of a transition in a CPN may lead to different markings according to the binding of the variables in the arc expressions. This means that the transition with its current binding identifies the firing step. A *firing step* is defined as a non-empty and finite multiset over the set of all (t, b) pairs, where t is a transition and b is a binding of variables in t. A firing step is usually denoted by Y. The sequence of firing steps is denoted by $M_0[Y_0 > M_1[Y_1 > \ldots [Y_{k-1} > M_k$.

Timing Colored Petri nets can be extended with the timing of the transitions. Time delays can be assigned to the transitions that affect the execution time of the transition. In timed CPNs the tokens have time stamps too, which denotes the time when the token becomes available. For example if a token has time stamp T_1 it is not available until the simulation time is less than T_1. The firing of a transition increments the time stamp of the tokens in the current binding with the execution time of the transition.

Occurrence graph The behavioural analysis of a CPN can be done with the help of the occurrence graph [103]. The occurrence graph contains all of the reachable markings (system states) from the given initial marking in a form of a directed graph. The nodes of the graph refer to the colored token distribution of the places at the current state of the CPN model. The edges of the occurrence graph refer to a (transition, binding) pair that takes the given marking to the next marking, i.e. there is an edge between two nodes with markings M_i and M_j, if there exists a firing step Y such that $M_i[Y > M_j$.

The occurrence graph of a CPN can be constructed similarly to the ordinary Petri nets. The difference is that the nodes are colored markings and the edge labels are (t, b) transition-binding pairs.

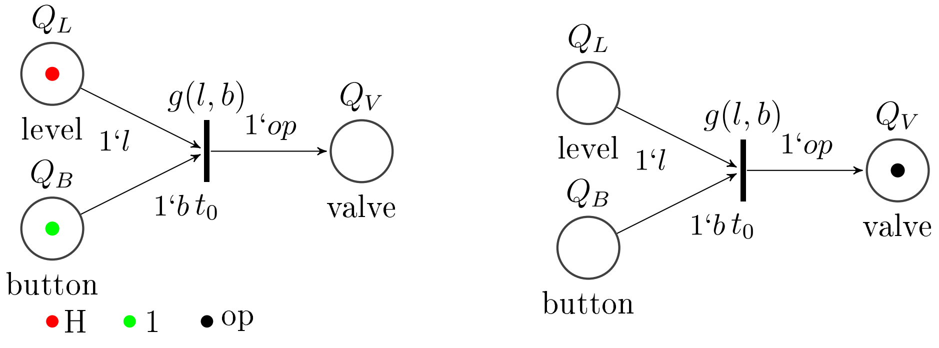
Simple example The operation of colored Petri nets is illustrated on a simple example that can be seen in Figure 4.2. The modelled process is the emptying of a water tank that has the following steps. If the water level in the tank is not empty then it may be emptied. To open the output valve (V_A) of the tank a button need to pressed to start the process. The places of the CPN model are *level*, *button* and *valve*. The corresponding color sets that are shown above the places are the following:

- $Q_L = \{0, L, N, H\}$ represents the level of the water in the tank (zero, low, normal, high);
- $Q_B = \{0, 1\}$ represents the state of the button (0 - not pressed, 1 - pressed);
- $Q_V = \{op, cl\}$ represents the state of the valve (open, close).

The transition t_0 represents the opening of the output valve. The guard function of the transition is

$$g(l, b) = \begin{cases} true, & \text{if } l \neq 0 \wedge b == 1 \\ false, & \text{otherwise} \end{cases}$$

where l and b are two variables with $Type(l) = Q_L$ and $Type(b) = Q_B$. The arc



(a) The initial state of the CPN

(b) The state of the CPN after firing t_0

$$\begin{tikzcd}
M_0 = [1`H, 1`1, empty]^T \arrow[d, "(t_0, < l = H, b = 1 >)"] \\
M_1 = [empty, empty, 1`op]^T
\end{tikzcd}$$

(c) The occurrence graph of the CPN

Figure 4.2: A simple colored Petri net and its occurrence graph

expressions are like edge weights in this case: $1`l$ means that one token with the color from $Q_L$ should be moved from *level*. $1`op$ means that one token with color *op* is moved to *valve*.

Initially there is one token on *level* with the color H and a tokens on *button* with color 1, as can be seen in Figure 4.2a. It can be seen that the binding $< l = H, b = 1 >$ makes the guard function true, therefore the transition is enabled in this case. During the firing of t_0 one token with color H is removed from *level* and one token with color 1 is removed from *button*. The transition puts one token with color *op* in place *valve*. The resulted state of the CPN can be seen in Figure 4.2b. The occurrence graph of this simple example can be seen in Figure 4.2c. The markings are represented by the list of places and the tokens on them.

4.2 Colored Petri net model of process systems

In this section the modelling methodology that can be used for creating CPN models of process systems is given. The system to be modelled is given with the set of traces that describes the different operational modes.

The role of the CPN model is twofold. First of all, it is able to simulate the normal and the different faulty operational modes of the system. On the other hand, the model also generates the deviations between the nominal and characteristic traces, which is the basis of our diagnostic method.

The inputs of the model are the nominal and characteristic traces. The outputs of the model are the qualitative values of the variables and the generated deviations between the nominal and the simulated characteristic trace at a given time. The diagnosis of the model can be done by the occurrence graph, that can be automatically generated from the initial state.

4.2.1 Modelling assumptions

The basic modelling assumptions are the following.

- The nominal and the characteristic traces describing the nominal and different faulty operations of the process need to be known in advance.
- The faults possibly occurring in the process are known in advance. Only these kinds of faults are included in the CPN model.
- In the simplest case only one fault occurs in the process.
- The occurring fault is present from the start of the process.
- The fault does not change during the process.

4.2.2 The structure of the CPN model

The CPN model that can be used for diagnosis of technological systems is introduced here. The general structure of the model can be seen in Figure 4.3.

In order to use the CPN model for diagnostic purposes the following modelling principles were used.

- The model simulates the operation of the technological system based on the known nominal and characteristic traces.
- The input and output variables, the operational mode, the deviations and the events of the nominal trace are assigned to places. The places are represented by ellipses in Figure 4.3.
- Colors represent qualitative values of the variables, the type of the fault and the emergent deviation from the nominal trace.
- A token with a given color at a place represents the actual qualitative value of the variable.
- The transitions execute the timing of the system. The transitions are represented by rectangles in Figure 4.3. The execution time of each transition is equal to 1.
- The enabling of the transitions is controlled by the simulation time.
- The fault generation function is attached to the arc pointing from t_1 to $fault$, and it is evaluated when the transition fires.

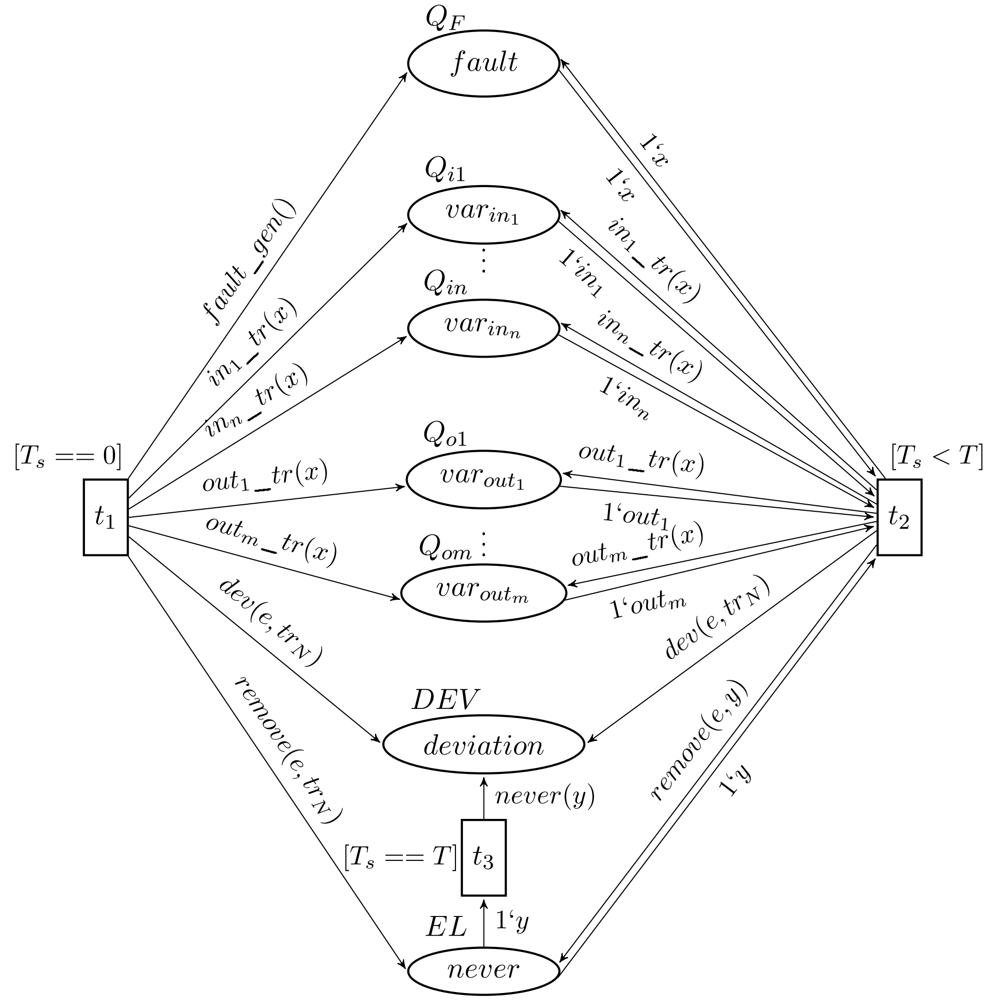


Figure 4.3: The structure of the CPN model

- The arc functions describe the change of qualitative values of the variables.

The declarations of color sets used in the model are the following:

- $Q_F = \{norm, fault_1, \ldots, fault_f\}$ is the color set of the possible operational modes, where $norm$ is the normal operation and $fault_1, \ldots, fault_f$ are the possible known faults;

- $Q_{ik} = \{q_{ik}^1, \ldots, q_{ik}^{j_k}\}$ is the color set of the kth input variable with the qualitative values $q_{ik}^1, \ldots, q_{ik}^{j_k}$, $k = 1, \ldots, n$;

- $Q_{ol} = \{q_{ol}^1, \ldots, q_{ol}^{j_l}\}$ is the color set of the lth output variable with the qualitative values $q_{ol}^1, \ldots, q_{ol}^{j_l}$, $l = 1, \ldots, m$;

- INT is the color set of integers;

- $E = INT \times Q_{i1} \times \ldots \times Q_{in} \times Q_{o1} \times \ldots \times Q_{om}$ is the color set of events;

- $KW = \{NH, LAT, EAR, SML, GRE\}$ is the color set of keywords used in the deviations;
- $DEV = KW \times E$ is the color set of deviations;
- $EL = list\ E$ is the color set of event lists (traces).

The declarations of variables that are used in the arc expressions are the following:

- $x \in Q_F$;
- $in_k \in Q_{ik},\ k = 1, \ldots, n$;
- $out_l \in Q_{ol},\ l = 1, \ldots, m$;
- $e \in E$;
- $y \in EL$;
- $T \in INT$.

The nominal and the characteristic traces are declared as constants:

- $tr_N = [event_N^1, event_N^2, \ldots, event_N^T]$ is the nominal trace of the process;
- $tr_{F_i} = [event_{F_i}^1, event_{F_i}^2, \ldots, event_{F_i}^{T_i}]$ is the characteristic trace of the ith faulty operational mode, $i = 1, \ldots, f$.

The descriptions of the used arc functions are the following:

- $fault\_gen() = \{$ let mode=discrete(0,f:INT);
  case $mode = 0 \rightarrow norm$;
  $mode = 1 \rightarrow fault_1$;
  $\vdots$
  $mode = f \rightarrow fault_f\}$
  is the fault generation function that randomly generates the operational modes with discrete uniform distribution;
- $in_k\_tr(x : Q_F)$ is the function that determines the value of the kth input variable ($k = 1, \ldots, n$) at the current time step from the trace of the corresponding operational mode x;
- $out_l\_tr(x : Q_F)$ is the function that determines the value of the lth output variable ($l = 1, \ldots, m$) at the current time step from the trace of the corresponding operational mode x;
- $remove(e : E, y : EL)$ removes the event e from the list of events y;
- $dev(e : E, y : EL)$ generates the EAR, LAT, GRE, SML type deviations between the current event e and the trace y;
- $never(y : EL)$ generates the NH type deviations by attaching the NH keyword to the event is in the list y;
- $tr\_length(x : Q_F)$ determines the length of the trace corresponding to the operational mode.

4.2.3 The operation of the CPN model

The aim of the CPN model is to simulate the technological process and generate the deviations between the nominal and the characteristic trace at the same time.

The operation of the previously introduced general CPN model can be divided into three sections based on the simulation time. The simulation time is denoted by T_s in Figure 4.3.

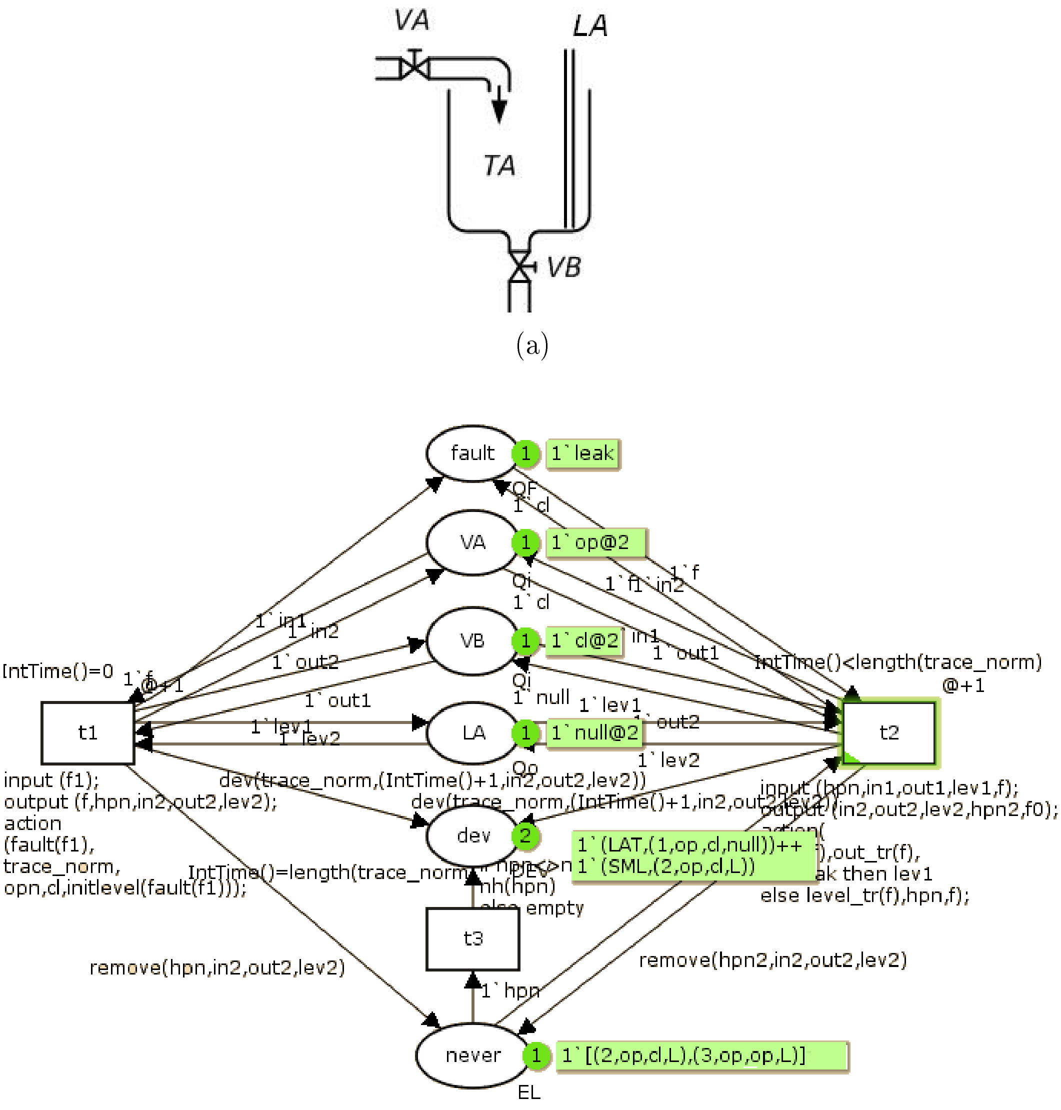
At the beginning of the simulation ($(T_s = 0)$ the current operational mode is generated and the variables are updated according to it. The only transition that is enabled at $T_s = 0$ is t_1. The firing of t_1 has the following effects:

- The operational mode (x) is generated using the function $fault\_gen$ and the token with the color of the operational mode is put in place $fault$.
- The end time of the simulation is determined as the length of the characteristic trace of the operational mode: $T = tr\_length(x)$.
- According to the operational mode, the values of the input and output variables at the first time step are put in places $var_{in_1}, \ldots, var_{in_n}$, $var_{out_1}, \ldots, var_{out_m}$ using the functions in_k_tr and out_l_tr.
- If there is a deviation between the nominal trace and the first event $e = (1, var_{in_1}(1), \ldots, var_{in_n}(1), var_{out_1}(1), \ldots, var_{out_m}(1))$ then token(s) with the deviation(s) are put in place *deviation*. The nominal trace is declared as a constant (see Section 4.2.2).
- The event is removed from the nominal trace (if possible) and the remaining part of the nominal trace is put in place *never* by the function *remove*.
- The simulation time is increased by 1.

The next time section interval lasts until the simulation time reaches the end time: $T_s < T_{end}$. During this time the steps of technological process are simulated. In this interval only t_2 is enabled and fires several times in a row. The effects of this transition are:

- The operational mode does not change in place $fault$.
- The values of the input and output variables are updated according to the operational mode and the current time.
- The deviations between the current event and the nominal trace are generated at each time step and the tokens with the proper colors are put in place *deviation*.
- The events that occurred in the nominal trace are removed from it and the remaining trace are put in place *never*.
- The simulation time is increased by 1 after each firing of t_2.

At the end of the process ($T_s = T_{end}$) the events that never happened remained in the place *never*. At this time the never happened type deviations can be generated by t_3. If there is any token in place *never* then t_3 can fire and attaches the NH keyword to the events in *never* and put these deviations in place *deviation*.


VA
LA
TA
VB
fault
1`leak
QF
1`cl
VA
1`op@2
Qi
1`cl
1`f1`in2
1`f
VB
1`cl@2
in1
1`out1
Qi
1`null
1`lev1
1`out2
LA
1`null@2
Qo
1`lev2
IntTime()=0
1`f
@+1
1`in1
1`in2
1`out2
1`out1
1`lev1
1`lev2
t1
input (f1);
output (f,hpn,in2,out2,lev2);
action
(fault(f1),
trace_norm,
opn,cl,initlevel(fault(f1)));
dev(trace_norm,(IntTime()+1,in2,out2,lev2))
dev(trace_norm,(IntTime()+1,in2,out2,lev2))
IntTime()<length(trace_norm)
@+1
t2
input (hpn,in1,out1,lev1,f);
output (in2,out2,lev2,hpn2,f0);
action(
),out_tr(f),
ak then lev1
else level_tr(f),hpn,f);
dev
2
1`(LAT,(1,op,cl,null))++
1`(SML,(2,op,cl,L))
DEV
IntTime()=length(trace_norm)
nh(hpn)
empty
t3
remove(hpn,in2,out2,lev2)
remove(hpn2,in2,out2,lev2)
1`hpn
never
1`[(2,op,cl,L),(3,op,op,L)]
EL

(a)

(b)

Figure 4.4: The tank and its CPN model of during the simulation ($T_s = 2$)

Simple example In order to better understand the operation of the CPN model a simple example is introduced. The modelled system is a tank with an input and output valve (VA and VB) and a level sensor (LA) (see in Figure 4.4a). The qualitative range values of the level sensor and the valves are Q_S and Q_B that were introduced in Section 4.1.1. The technological process is the filling of the tank. The nominal trace of the process is $[(1, op, cl, 0), (2, op, cl, L), (3, op, op, N)]$. It is assumed that only one fault may occur in the system that is the leakage of the tank. The characteristic trace of the leakage is $[(1, op, cl, 0), (2, op, cl, 0), (3, op, op, 0)]$.

The CPN model of the tank can be seen in Figure 4.4b.

The model operates in the following way. At $T_s = 0$ the current operational mode (normal or leak) is generated by t_1 and the result is put in place $fault$. Figure 4.4b represents that case when the tank has a leak. It can be seen that there is a token with color *leak* in place $fault$. The end time of the simulation (the length of the characteristic trace) is $T = 3$. The values of the valves and the level sensor are also set to *op*, *cl* and 0 at $T_s = 0$. At this time there is no deviation between the nominal and the characteristic trace.

In the next step transition $t2$ fires while $T_s < T_{end}$. During this time the values of the valves and the level sensor are updated according to the characteristic trace of leakage. Besides that the deviations are generated at every time step. The current state of the CPN model at $T_s = 2$ can be seen in Figure 4.4b. It can be seen that place VA, VB and LA has tokens with color *op*, *cl* and 0. The deviations generated at $T_s = 2$ are $LAT(1, op, cl, 0)$ (because the current measured event is a later occurrence of the first nominal event) and $SML(2, op, cl, L)$ (because the current level is smaller than in the second nominal event). The deviations can be found in place *dev.*. The never happened events are $(2, op, cl, L)$ and $(3, op, op, N)$.

The process ends when the simulation time reaches $T_s = 3$. A new quantitative deviation is generated, which is $SML(3, op, op, N)$. At this time the never happened type deviations are generated too, which are $NH(2, op, cl, L)$ and $NH(3, op, op, N)$.

4.2.4 Generating the occurrence graph

The CPN Tools software was used for the modelling, simulation and analysis of colored Petri nets [104]. To generate the occurrence graph of the model that contains all possible states of the system in case of the known operational modes, the nominal and the characteristic traces of different faults are need to be known. The nodes of the graph contain the colored token distribution (i.e. the number and color of tokens) in the places of the CPN model. From diagnostic point of view the following attributes of the nodes are important:

- Token in place $fault$. This is the operation mode of the process.
- Tokens on input and output variable places. These are the qualitative values of the variables at the current state.
- Tokens in place *deviation*. This is the deviation list between the nominal and the characteristic trace of a known fault at the current time of the process.

In the general case the occurrence graph of a colored Petri net may contain cycles and multiple directed paths between two nodes. Fortunately in our case the occurrence graph has *tree structure*, which follows from the special construction of the CPN model. The root of the tree is the initial state and each branch of the tree belongs to one operational mode.

4.2.5 Multiple faults, faults on the fly

In a real process system faults can occur at any time and not only one fault can influence the course of the system. To fulfil these requirements the basic model can be modified in the following way.

For the management of the effect of two or more faults at the same time new fault types were introduced. These new types were added to the list of possible faults, so the adequate faulty operational mode can be generated at the initialization of the system. The characteristic traces of these faulty modes also need to be known and added to the set of traces.

To model the fault occurring at any time it is assumed that the system works in normal way until this point of time. It is assumed that the fault does not change until the end of course of the system. Using this assumption the faulty operational mode has to be modelled from this point of time. For the diagnosis the traces describing the events from this step should be generated and added to the model. The additional modelling assumptions are the following.

- The fault occurs at a *known* time τ.
- The system operates in normal mode until τ.
- The course of the system is modelled from τ.

4.3 Unit-wise diagnosis

In this section the basic diagnostic method of one technological unit is presented. The basic principles introduced here can be later adapted to the case of diagnosis of composite systems.

4.3.1 Diagnosis using the occurrence graph

The basic idea of the diagnosis is that if a fault occurred in the system, then there are deviations between the nominal and the measured trace. In case of known faults, the deviations between the nominal and the characteristic traces are also known, and the occurred fault can be identified from the deviations. If a measured trace from the system is available, then the deviations from the nominal trace can be generated. These deviations can be compared to the known deviations between the nominal and characteristic traces. If the two deviations match then the occurred fault is that one whose characteristic trace caused that deviations.

In the proposed diagnostic method the occurrence graph of the CPN model is used to store the deviations between the nominal and the known characteristic traces. The diagnostic method can be realized as a searching on the occurrence graph.

Let us assume that all the fault modes and the corresponding characteristic traces of the modelled technological system are known and a measured trace is available from the process. The steps of the diagnosis using the occurrence graph are the following.

1. Generate the occurrence graph of the CPN model of the system. The resulted occurrence graph contains all of the possible system states reachable from the initial state. This step can be done off-line.

2. Generate the deviation list between the nominal and the measured trace.

3. Find the node having the same deviation list in the place *deviation* as it was generated in the previous step. After finding that node, the occurred fault can be determined based on the token color on place $fault$ at this node. There may be three cases:

   - There is only one node with the same deviations as the generated deviation list. Then the occurred fault is clearly identified.
   - There are more than one node with the same deviations as the generated deviation list. Then only the set of possible faults can be identified.
   - If no token distribution refers to the generated deviation list then an unknown fault occurred in the system. In this case the set of diagnosed faults is empty, but the deviation list is not empty. This combination indicates the presence of an unknown fault, which can be *detected but not identified.*

The diagnosis can be done on-line or off-line. In case of on-line diagnosis the full measured trace is not known, but a new event occurs at each time step. In this case steps 2-3 are repeated at every time step. This means that the deviations between the nominal and measured traces are updated after each new event, and the resulted deviations are searched on the occurrence graph.

In case of off-line diagnosis, the full measured trace is known. Therefore the deviations need to be generated once and they can be searched between the terminal nodes of the occurrence graph.

4.3.2 The diagnostic algorithm

The formal description of the steps of the diagnosis of one technological unit are presented here. At first the algorithm that generates the deviations between two traces is introduced. Deviations between traces are used in HAZID based methodologies [105]. In Algorithm 3 a different version of the algorithm generating deviations is presented using the notations of colored Petri nets.

Algorithm 3 Generating deviations between two traces

```
Input:
$tr_N = [event_N^1, event_N^2, \ldots, event_N^T]$                    // nominal trace
$event_N^t = (t, var_{in_1}^N(t), \ldots, var_{in_n}^N(t), var_{out_1}^N(t), \ldots, var_{out_m}^N(t))$
                                       // event at time $t$ in the nominal trace
$tr_C = [event_C^1, event_C^2, \ldots, event_C^{T_C}]$          // measured/characteristic trace
$event_M^t = (t, var_{in_1}^C(t), \ldots, var_{in_n}^C(t), var_{out_1}^C(t), \ldots, var_{out_m}^C(t))$
                        // event at time $t$ in the measured/characteristic trace
function DevGen($tr_C, tr_N, T$)
    $D = \emptyset$                            // multiset of deviations (initially empty)
    $neverlist = tr_N$                              // list of never happened events
    $T_C$=length($tr_C$)
    $T_N$=length($tr_N$)
    for $t = 1$ to $T$ do
        if $event_C^t == event_N^t$ then
            remove($neverlist, event_N^t$)
            //quantitative deviations:
        else if $(var_{in_i}^C(t) == var_{in_i}^N,\ i = 1, \ldots, n) \wedge (\exists\ 1 \leq j \leq m :$
            $(var_{out_j}^C(t) < var_{out_j}^N)$ then
            $D = D + 1`SML_j(event_N^t)$
        else if $(var_{in_i}^C(t) == var_{in_i}^N,\ i = 1, \ldots, n) \wedge (\exists\ 1 \leq j \leq m :$
            $(var_{out_j}^C(t) > var_{out_j}^N)$ then
            $D = D + 1`GRE_j(event_N^t))$
            //chronological deviations:
        else if $\exists \tau < t : (var_{in_i}^C(t) == var_{in_i}^N(\tau)) \wedge (var_{out_j}^C(t) == var_{out_j}^N(\tau))$
            $i = 1, \ldots, n,\ j = 1, \ldots, m$ then
            $D = D + 1`LAT(event_N^t))$
            remove($neverlist, event_N^t$)
        else if $\exists \tau > t : (var_{in_i}^C(t) == var_{in_i}^N(\tau)) \wedge (var_{out_j}^C(t) == var_{out_j}^N(\tau))$
            $i = 1, \ldots, n,\ j = 1, \ldots, m$ then
            $D = D + 1`EAR(event_N^t))$
            remove($neverlist, event_N^t$)
        end if
    end for
    if $T == max(T_N, T_C)$ and $neverlist \neq [\,]$ then
        //never happened type deviations:
        for $i = 1$ to length($neverlist$) do
            $D = D + 1`NH(neverlist(i)))$
        end for
    end if
return $D$
end function
```

Algorithm 3 describes the process of generating the deviations between two traces: the measured or the characteristic trace (tr_C) and the nominal trace (tr_N) up to time T for on-line diagnosis. The inputs of the DevGen function are the measured/characteristic and the nominal traces and the time T up to that the deviations are generated. The output is the multiset of deviations D.

The algorithm goes through the events of the measured/characteristic trace and compare them with the events of the nominal trace. At first the quantitative deviations (SML, GRE) are generated and added to the multiset D. After that the chronological deviations (EAR, LAT) are generated and added to D. The events that implied chronological deviations are removed from the list of never happened events. At the end of the algorithm the never happened type deviations are created.

The diagnosis of a technological unit is done by searching the nodes of the occurrence graph, whose marking in place *deviation* matches the deviations between the measured and nominal traces. Algorithm 4 describes this diagnostic process.

The inputs of the DiagUnit function are the deviations to be found (D), the occurrence graph ($G = (V, A)$) of the CPN model and the current time T. The output of the model is the multiset of occurred faults F.

During the diagnosis it is enough to perform the search between the nodes with the time stamp T, which are in the set N. In the algorithm we go through the nodes in N and compare the marking in place *deviation* with the current deviations D. The marking in place *deviation* at the ith node is denoted by $M_i(deviation)$. If the two deviations are equal then the marking in place $fault$ of the ith node $M_i(fault)$ is added to the set of faults F.

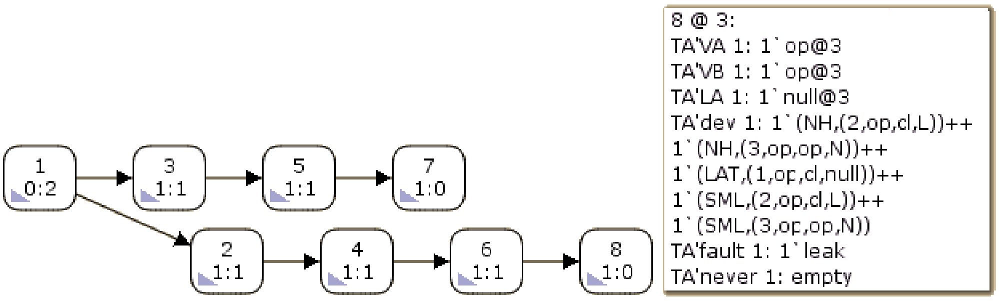
Algorithm 4 Diagnosis of a technological unit

```
Input:
$D = DevGen(tr_c, tr_N, T)$                          // multiset of deviations
$G = (V, A)$                                         // occurrence graph
function DiagUnit($D, G, T$)
    $F = \emptyset$              // multiset of occurred faults (initially empty)
    $N = \{v \in V : time(v) == T\}$
                          // node with time stamp $T$ of the occurrence graph
    for $i = 1$ to $|N|$ do
        if $M_i(deviation) == D$ then
            $F = F + M_i(fault)$
        end if
    end for
return $F$
end function
```

Simple example The unit-wise diagnostic method is illustrated on a simple example. The examined system is the tank with a possible leak that was introduced in Section 4.2.3. The occurrence graph of the CPN model can be seen in Figure 4.5. It has two branches referring to the normal operation or the leakage. Let us assume that the measured trace is $[(1, op, cl, 0), (2, op, cl, 0), (3, op, op, 0)]$ and the process has ended, i.e. the current time is $T = 3$. At first the deviations (D) between the nominal and the measured trace until $T = 3$ are generated. The deviations can be seen below.

$$\begin{aligned} D = {} & LAT(1, op, cl, 0), SML(2, op, cl, L), SML(3, op, op, N), \\ & NH(2, op, cl, L), NH(3, op, op, N) \end{aligned}$$

Then this deviation list is searched among the nodes of the occurrence graph. Node No. 8 was found to have the same deviations in place *dev* (i.e. the marking of this place is $M_8(dev) = D$). The token distribution of node No. 8 can be seen on the right of Figure 4.5. It can be seen that this node has a token with color *leak* in place $fault$ (i.e. $M_8(fault) = leak$), therefore the occurred fault in the system is $F = \{leak\}$.



```
8 @ 3:
TA'VA 1: 1`op@3
TA'VB 1: 1`op@3
TA'LA 1: 1`null@3
TA'dev 1: 1`(NH,(2,op,cl,L))++
1`(NH,(3,op,op,N))++
1`(LAT,(1,op,cl,null))++
1`(SML,(2,op,cl,L))++
1`(SML,(3,op,op,N))
TA'fault 1: 1`leak
TA'never 1: empty
```

Figure 4.5: Occurrence graph of the tank

4.4 Diagnosis with structural decomposition

The disadvantage of occurrence graph based method is the increasing size of the graph with the increase of the size (i.e. number of places) of the CPN model. Even in case of a simpler technological system, which contains three or four units the occurrence graph of its CPN model can contain hundreds of nodes depending on the number of sensors and actuators. The refinement of the qualitative measuring range of sensors or the application of control valves instead of two-state actuators may also cause the growing of occurrence graphs because their branches will be longer. With the growth of the size of the graph, the computational effort and searching-time also increases. This is the reason why the structural decomposition has crucial impact on the practical application of the diagnostic process.

4.4.1 Composite systems

A composite system is composed of more than one process units, which are connected to each other. The structure of the system and the connections between the individual units are usually known from technology diagrams. In a composite system the units may connect to each other in parallel or sequential structures. A unit may have one or more direct preceding or subsequent units. In a composite system the number of possible faults can be much larger than in an individual unit, and diagnosing them is a challenging task. The concepts of faults in composite systems and their modelling are presented below.

Faults in composite systems

The main problem with the diagnosis of composite systems is that faults may occur in each individual unit simultaneously and there may be faults that affect the operation of other units. Each unit has its own fault types that contributes the fault of the whole system. The global fault of the composite system is originated from one or more of its units. However there may be faults that exclude each other the number of possible combinations of faults occurring in the units may be extremely large. To simulate all possible faults and create the occurrence graph of the CPN model of the system, all traces describing the effect of possible faults should be known in advance. During the diagnosis of a composite system, both the location and the type of the fault should be determined.

CPN model of composite systems

Instead of creating a global CPN model of the composite system, the system was cut along the connection points of its units and the units of the system are modelled one by one with individual CPNs. The CPN model of each unit has the same structure as it was introduced in Section 4.2.

To take into account the effect of faults occurred in previous units, the color set of faults was modified. Let us assume that the composite system has U number of units and each unit has j_i number of faults. The new color set contains both the faults and the identifier of the unit where the fault occurred. Formally this color set is the product of faults and unit identifiers:

$$\begin{aligned}
Q_{F_i} &= \{norm, fault_1^i, fault_2^i, \dots, fault_{j_i}^i\},\ i = 1, \dots, U \\
ID &= \{id_1, id_2, \dots, id_U\} \\
Q_{F,ID} &= \{(fault^i, id_i) | fault^i \in Q_{F_i}, id_i \in ID, i = 1, \dots, U\}
\end{aligned}$$

where Q_{F_i} is the set of fault in the ith unit, ID is the set of unit identifiers, and $Q_{F,ID}$ is the set of faults with unit identifiers.

Let us consider one technological unit. If a fault has occurred in a preceding unit with identifier id_i then this fault with the identifier of that unit is added to the place $fault$ in the CPN model of the actual unit, as an initial condition. For example if the unit with id_i is the preceding unit of the actual unit and a

$fault^i_j$ has occurred in it, then the token with color $1`(fault^i_j, id_i)$ is added to place $fault$. After that the actual fault in the unit can be generated and the unit can be simulated taking into account the traces referring to the operation mode of the faults in the current and previous units in the system.

4.4.2 Structural decomposition

As it was already mentioned, in a composite process system the structure and the connections of components are usually known. Knowing the structure of the system it seems evident to diagnose the system using structural decomposition. Diagnosis with structural decomposition means that the composite system is partitioned into basic individual units along the connection points and then the diagnosis is performed on them separately. The final global diagnostic result can be concluded from the results of the unit-wise diagnosis. Knowing the technological sequence of the units the diagnosis should be started with the first unit and its result has to be taken into account when diagnosing the connected units.

The advantage of the decomposition is that the diagnosis of the units is usually simpler and faster than a global diagnostic procedure. Moreover the previously introduced unit-wise diagnostic method (see Section 4.3.2) can be easily adapted to the decomposed system.

Decomposition of traces

The events in the global trace contain all variables of the system. However the operation of a unit can be described with only the variables corresponding to the given unit. For the structural decomposition based diagnostic method the full trace of the system should be decomposed too by selecting the variables and time instances that describe the operation of single units. The input and output variables of a unit can be determined from the technology diagram. Moreover it may occur that a unit does not operate during the whole process (e.g. in case of a sequential process). Therefore the starting and finishing time in between the unit operates also need to be determined. In the subtrace of the unit relative time is used in such a way that the real time is shifted back to 1. So in each subtrace belonging to a given unit, the operation of the given unit starts at time step 1. The decomposition of the global trace is presented in Algorithm 5. Let us assume that the composite system has U number of units. The function UnitTrace extracts one subtrace of a unit tr_u with variables in Var_{unit} and with start time T^S and finish time T^F. This function is called for all units of the composite system with the corresponding set of variables Var_{unit_i} and times T^S_i, T^F_i. The result of the algorithm is all subtraces corresponding to separate units.

Algorithm 5 Decomposition of the global trace

```
$tr = (event^1, event^2, \dots, event^T)$
$Var_{unit} = \{var_{in_{i1}}, \dots, var_{in_{ik}}, var_{out_{j1}}, \dots, var_{out_{jl}}\}$
                                        // set of variables corresponding to the unit
function UnitTrace($tr, Var_{unit}, T^S, T^F$)
    $tr_u = [\,]$
    $t = 1$
    for $i = T^S$ to $T^F$ do
        $event_u^t = (t, var_{in_{i1}}(i), \dots, var_{in_{ik}}(i), var_{out_{j1}}(i), \dots, var_{out_{jl}}(i))$
        $tr_u = [tr_u, event_u^t]$
        $t = t + 1$
    end for
return $tr_u$
end function
for $i = 1$ to $U$ do
    $tr_{u_i} =$ UnitTrace($tr, Var_{unit_i}, T_i^S, T_i^F$)
end for
```

Diagnostic algorithm

The diagnosis of the composite system is carried out in the technological order of the units. The units are diagnosed one by one and the diagnosed faults of a unit are taking into account at the following units.

To generate deviations corresponding to the examined unit, the nominal and the measured trace should be decomposed first. Then the deviations between the nominal and measured subtraces can be computed.

The diagnostic algorithm of a unit is similar to the unit-wise diagnosis (Algorithm 4 in Section 4.3.2). The difference is that the faults occurred in previous units can be taken into account at the generation of the occurrence graph. As it was introduced in Section 4.4.1 the previously occurred faults can be added as initial conditions to the CPN model. If a unit has more preceding units (e.g. in a parallel structure), then the initial condition contains the diagnosed faults in all previous units. Considering the initial conditions, the occurrence graph of the unit can be generated that contains all reachable states *assuming the occurred faults in previous units.* The deviation lists of the nodes are the deviations between the nominal and characteristic subtraces. To generate the deviations, the characteristic subtraces of the possible fault combinations need to be known, too.

The diagnostic algorithm of composite systems is presented in Algorithm 6. It is assumed that the units are ordered according to the technological order, i.e. $unit_i$ precedes $unit_{i+1}$ in the technological order. The parallel units can be processed simultaneously in one diagnostic step.

The function DiagComp contains the diagnostic algorithm that uses the previously introduced UnitTrace, DevGen and DiagUnit functions, too. The diagnosed faults are collected in set F as $(fault, unit)$ pairs, which represent the diagnosed fault and the unit where is occurred.

Algorithm 6 Diagnosis of a composite system

```
$tr_N = (event_N^1, event_N^2, \ldots, event_N^T)$    // nominal trace
$tr_M = (event_M^1, event_M^2, \ldots, event_M^{T_M})$    // measured trace
$Var = \{Var_{unit_1}, \ldots, Var_{unit_U}\}$    // set of variable sets of the units
$T^S = [T_1^S, \ldots, T_U^S]$    // start times of the operation of the units
$T^F = [T_1^F, \ldots, T_U^F]$    // finish times of the operation of the units
function DiagComp($tr_N, tr_M, Var, T^S, T^F, T$)
    $F = \emptyset$    // set of diagnosed faults
    for $i = 1$ to $U$ do
        if $T - T_i^S > 0$ then
            $F_i = \emptyset$    // set of diagnosed faults in $unit_i$
            $tr_N^{u_i}$=UnitTrace($tr_N, Var_{unit_i}, T_i^S, T_i^F$)
            $tr_M^{u_i}$=UnitTrace($tr_M, Var_{unit_i}, T_i^S, T_i^F$)
            if $T_i^F \leq T$ then
                $T^D = T - T_i^S$
            else$T^D$ =length($tr_M^{u_i}$)
            end if
            $D_i$=DevGen($tr_M^{u_i}, tr_N^{u_i}, T^D$)
            for all $pf \in F_1 \times \ldots \times F_{i-1}$ do    // previous fault combinations
                in the CPN model of $unit_i$ add $pf$ to place $fault$
                $G_i = (V, A)$    // occurrence graph of $unit_i$
                $f$=(DiagUnit($D_i, G_i, T^D$),$unit_i$)
                $F_i = F_i \cup f$
            end for
        end if
        $F = F \cup F_i$
    end for
return $F$
end function
```

The main advantage of the described method is the smaller size of the occurrence graphs of subsystems therefore the searching requires less time than in case of the investigation of the whole technological system.

The structural decomposition based diagnostic method is explained on a simple case study that can be found in Appendix B. The examined technological system is composed of three tanks in serial connection. However the example is relatively simple the key features of the decomposition and the diagnostic algorithm can be well observed.

4.5 Discussion and future work

Diagnosis based on event sequences and deviations are used in this chapter for process system diagnosis. In this section two similar methods are compared with the method presented here. The first method is called *P-HAZID (Procedure HAZard IDentification)* and it uses backward reasoning in a P-HAZID table to find the root causes of the failures [105]. One row of a P-HAZID table contains the deviations with their causes and consequences. The second one is a *clustering diagnoser* that uses the distance of the nominal and characteristic traces to identify the occurred faults [106]. In this method the traces are converted into coordinate vectors, and the distance of the transformed traces is used to compare traces instead of deviations. The traces having similar distances are collected in clusters, as they most likely belong to the same fault. The similarities of these works and my diagnostic methods are the following:

- Qualitative event sequences are used to describe the processes.
- All methods are based on the comparison of nominal and characteristic traces.
- The same basic assumptions are made for the faults: the faults are permanent and present at the beginning of the diagnosis.
- All methods have similar diagnostic capabilities. The problem with unknown faults and not clearly identified faults are the same at the three methods.
- Faults that are dependent cannot be identified.
- The clustering diagnoser has an off-line training phase and an on-line diagnoser phase, similarly to my proposed method.

The main differences between the methods are:

- The P-HAZID method can be applied off-line, while my method is able to identify faults on-line.
- The P-HAZID method uses a recursive backward reasoning algorithm to obtain the root cause of the failure. In my method the fault mode is present in the node of the occurrence graph, therefore there is no need for reasoning.
- In the clustering diagnoser more slightly different traces may belong to the same fault. Therefore the disturbances have less effect on the efficiency of the clustering diagnoser. My method can diagnose faults in case of exact matching of the traces.

Advantages of my method:

- The nodes of the occurrence graph contains all important information about the current state of the process: values of variables, deviations and fault name.
- In contrast to the P-HAZID method, the occurred fault can be immediately identified from the found node with the proper deviation list.

- In the clustering diagnoser different faults (with different characteristic traces) may have the same distance from the nominal trace (e.g. opposite effects), which affects the diagnostic accuracy. In my method faults with different characteristic traces are always distinguished.

Future work In the presented work the examined technological processes were simple serial processes. In real technological systems the technological processes often contain loops or recirculation of materials (e.g. in chemical processes). The problem is that the traces of such processes can be slightly different, which makes the nominal/characteristic traces uncertain. Therefore the application of the developed diagnostic method in the present form could be problematic. The extension of the method will be investigated in case of such processes during the future work.

Another drawback of my approach is that the assumptions about the faults are very strict. In reality faults are not necessarily present at the beginning of the process but may occur at any time, which is usually not known in advance. A possible solution of this problem and that of the uncertainty in the traces could be the application of temporal logic [107],[108]. Using temporal logic the occurrence of events at some point of the process, and their consequences can be analyzed. With this approach the problem of time shifting of the events could be handled.

4.6 Summary

In this chapter a colored Petri net based modelling and diagnostic method of technological systems were presented. A general CPN modelling methodology was developed to represent the technological units of the system. In the CPN model qualitative range spaces were used for the characterization of the values of the variables. The model is able to simulate the normal and different faulty operational modes of the technological unit. The different operational modes are represented with their traces. The CPN model also generates the deviations between the nominal and the characteristic traces, which is essential from the diagnostic point of view.

The presented on-line diagnostic method is based on the occurrence graph of the CPN model. The basic idea of the diagnosis is that if a fault occurs then there are deviations between the nominal and the characteristic trace. The occurrence graph of the CPN model contains all possible reachable states of the system together with the deviations between the normal and the given faulty operational modes. Therefore the deviations can be searched on the graph and the fault can be identified from the token colors of the found node. The diagnostic algorithm applicable to a single technological unit is given in Algorithm 4.

In case of composite systems the diagnosis can be performed using structural decomposition. The units of the technological system are diagnosed one by one and the effects of faults in previous units are taken into account. As a result of the structural decomposition the diagnosis has to be performed on much smaller occurrence graphs. The diagnostic algorithm of composite systems is presented in Algorithm 6. This method could reduce the demand on computational effort and searching time.

An illustrative case study of the proposed method can be found in Appendix B. This simple case study illustrates the advantage of decomposition while the effect of faults on the diagnostic accuracy can also be observed.

Chapter 5

Theses

In this work different kinds of model based diagnostic methods with energetic applications were presented. The achieved new results are summarized in the following thesis points.

5.1 Thesis 1 - Parameter estimation method for temperature dependent battery parameters (Chapter 2)

Relevant publications: [**T1_17**], [**T1_18**], [**T1_19**]

The aim of this work was the estimation of the battery parameters relevant for the battery aging, especially the battery capacity. A special parameter estimation method for diagnostic purposes was developed which uses the simplest EECM model extended with temperature dependency [58].

T1.1 The temperature dependent battery model was analysed with respect to parameter sensitivity. The parameter sensitivity analysis showed that the model output is insensitive to the internal resistance while sensitive to the electrode potential, battery capacity and the two polarization constants. It was also shown that the battery capacity can be estimated from charging/discharging profiles.

T1.2 It was also found that the parameters can be estimated from the charging profiles only, which is essential from the application point of view as the charging profile of the battery can be influenced by the user.

T1.3 A two-step parameter estimation method was developed to estimate the temperature dependent parameters. In the first step of the method the parameters at different ambient temperatures are estimated. In the second step the characteristics of the temperature dependency are estimated using the results of the first step.

5.2 Thesis 2 - Non-technical loss diagnosis in electrical networks (Chapter 3)

Relevant publications: [**T2_18a**], [**T2_18b**], [**T2_19**]

In this work non-technical loss diagnosis in low-voltage electrical networks was investigated. The network was modelled by a simple static linear model. The structure of the network was represented by a directed graph. The proposed diagnostic method of electrical networks with complex structures is composed of two steps: decomposition and diagnosis of the network.

T2.1 A decomposition algorithm of complex networks was developed that decomposes the networks into a set of smaller basic one feeder subnetworks.

T2.1.a The decomposition of a radial one feeder network is based on cutting off the branches of the network. The resulted subnetworks may be single load subnetworks with only one load or multiple load subnetworks.

T2.1.b The decomposition of a two feeder network is done in parallel with the computation of the network variables. The result of the decomposition is two one feeder network. If the resulted network has radial layout then it can be further decomposed with the method in T2.1.a.

T2.2 A diagnostic algorithm that is able to detect and localize multiple illegal loads in electrical networks with different layouts was developed. The method takes into account the parameter uncertainties and the measurement errors in the network and also approximates the magnitude of the illegal current. The principle of the method is the analysis of the differences between the measured and the nominal values of the network. The measured and nominal currents are used for fault detection while the measured and nominal voltages are used for fault isolation.

T2.2.a The diagnosis of the radial one feeder network is performed on the decomposed network. If illegal loads are detected then the localization of them is done in two steps. At first illegal loads are localized in multiple load subnetworks and then in single load subnetworks. The localization within the subnetworks can be done in parallel.

T2.2.b The diagnosis of the two feeder network can be done with the converted basic two feeder network. The localization of the illegal loads is done by analyzing the computed illegal currents. If the illegal load is in a merged branch then the diagnostic method in T2.2.a can be applied to this branch.

5.3 Thesis 3 - Colored Petri net based diagnosis of technological systems (Chapter 4)

Relevant publications: **[T3_14]**, **[T3_16a]**, **[T3_16b]**

In this work a prediction based diagnostic method for technological systems was developed. The principle of the method is that the faults can be characterized by the deviations between the characteristic traces and the nominal trace if the traces of known faults are known. The technological units were modelled by colored Petri nets which is a dynamic discrete event model.

T3.1 A special colored Petri net (CPN) model was constructed that is able to simulate the nominal and different known faulty operations of the technological unit. The model also generates the deviations between the nominal and the characteristic traces. The occurrence graph of the CPN model is used for the diagnosis and it can be generated offline.

T3.2 The on-line diagnostic method is based on searching the deviations between the nominal and the measured trace on the occurrence graph of the CPN model. The occurred fault can be identified from the token colors of the found node. In case of composite systems the diagnosis is done by structural decomposition. The units of the system are diagnosed one by one taking into account the faults occurred in previous units.

5.4 Related publications

[T1_17] **A. I. Pózna**, A. Magyar, and K. M. Hangos: "Model identification and parameter estimation of lithium ion batteries for diagnostic purposes", in *2017 International Symposium on Power Electronics (Ee)*, IEEE, 2017, pp. 1-6. *Citations: 4*

[T1_18] **A. I. Pózna**, K. M. Hangos, and A. Magyar: "Design of experiments for battery aging estimation", *10th Symposium on Control of Power and Energy Systems (CPES2018)* in *IFAC-PapersOnLine*, vol. 51, no. 28, pp. 386-391, 2018. *Citations: 1*

[T1_19] **A. I. Pózna**, K. M. Hangos, and A. Magyar: "Temperature dependent parameter estimation of electrical vehicle batteries", *Energies*, vol. 12, no. 19, p. 3755, 2019, **IF=2.707**.

[T2_18a] **A. I. Pózna**, A. Fodor, and K. Hangos: "Non-technical loss diagnosis in electrical networks with a radial layout", *Hungarian Journal of Industry and Chemistry*, vol. 47, no. 1, pp. 3-9, 2018.

[T2_18b] **A. I. Pózna**, A. Fodor, M. Gerzson, and K. Hangos: "Colored Petri net model of electrical networks for diagnostic purposes", *IFAC-PapersOnLine*, vol. 51, no. 2, pp. 260-265, 2018. *Citations: 2*

[T2_19] **A. I. Pózna**, A. Fodor, and K. M. Hangos: "Model-based fault detection and isolation of non-technical losses in electrical networks", *Mathematical and Computer Modelling of Dynamical Systems*, pp. 1-32, 2019, **IF=0.862**.

[T3_16a] **A. I. Pózna**, M. Gerzson, A. Leitold, and K. Hangos: "Colored petri net based diagnosis of process systems", in *30th European Simulation and Modelling Conference - ESM'16*, J. J. Hernández-Cabrera, J. Évora-Gómez, Ed., European Multidisciplinary Society for Modelling and Simulation Technology (EUROSIS), 2016, pp. 91-95. *Citations: 3*

[T3_16b] **A. I. Pózna**, M. Gerzson, A. Leitold, and K. M. Hangos: "Diagnosis of technological systems based on the structural decomposition of their coloured petri net model", *Hungarian Journal of Industry and Chemistry*, vol. 44, no. 2, pp. 121-128, 2016.

[T3_14] A. Leitold, M. Gerzson, **A. I. Pózna**, and K. Hangos: "On-line qualitative model-based diagnosis of technological systems using colored Petri nets", in *28th European Simulation and Modelling Conference - ESM'2014*, A. C. Brito, J. M. Tavares, and C. B. de Olivera, Eds., EUROSETI-ETI, 2014, pp. 332-336.

Appendix

A Case study of model based diagnosis of electrical networks

The whole decomposition and diagnostic method is illustrated on a case study. The test network is the IEEE 2015 European Low Voltage Test Feeder [109], that has a radial structure. The simulation, decomposition and the diagnostic algorithm of the network were implemented in MATLAB [110].

A.1 Electrical network and its decomposition

The IEEE test network

The network is a three-phase network with one feeder and 55 loads. Phase A,B and C has 21, 19 and 15 loads respectively. The consumed power of each load is given with a load profile with one minute resolution. There are 905 transmission line segments and 906 buses in the network specification. The coordinates of the buses, the length, resistances of the lines and the specification of the transformer are given in separate csv files. The structure of the network can be seen in Figure A.1.

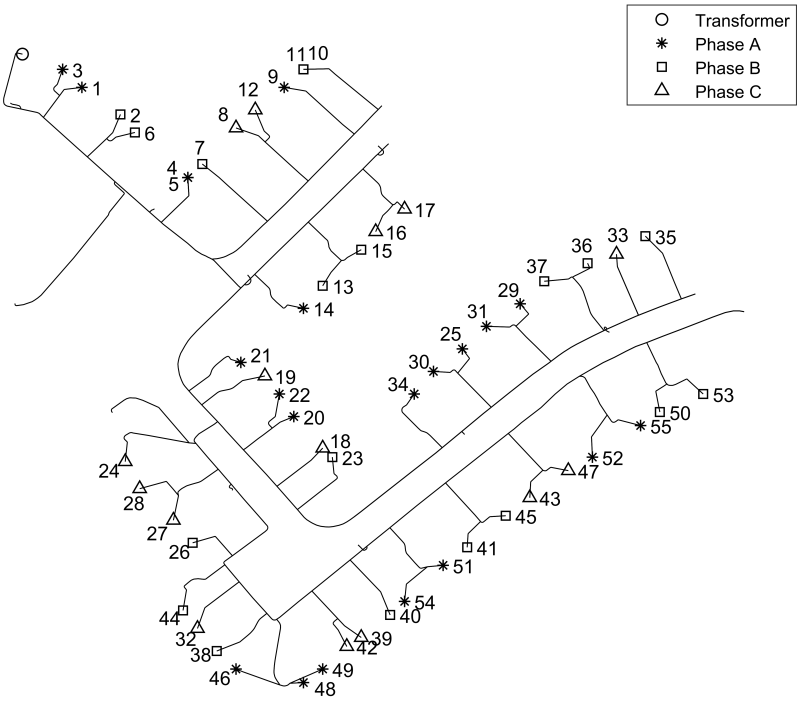


Figure A.1: The IEEE 2015 Low Voltage European Test Feeder network

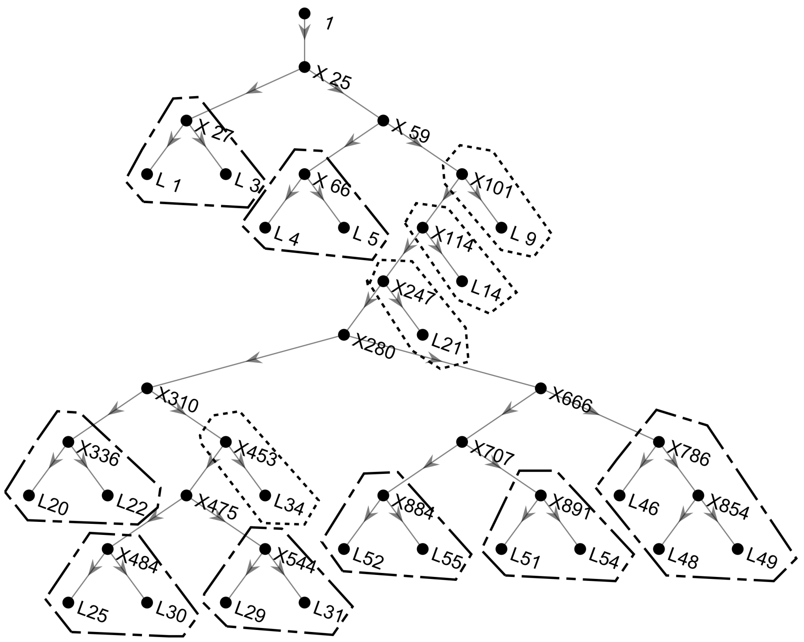


Figure A.2: The Phase A network of the IEEE 2015 Low Voltage European Test Feeder and its decomposition

To test the proposed diagnostic algorithm only one phase (Phase A) of the network was used. Moreover, the original network contains a lot of lines and buses, therefore it needs to be simplified so that the decomposition and diagnostic algorithms can be applied. The following steps are needed to convert the original network into a tree-graph:

- The loads that are not in Phase A are deleted.

- There are end nodes in the network that are not loads and they have no consumption data. These nodes are also deleted.

- The transmission lines are composed of several consecutive line segments defined by the connecting buses. Therefore a transmission line between two nodes is represented by consecutive edges in the original network. The line segments between two real nodes (load or junction point) need to be merged to get only one edge between two nodes. The resistance of the merged edge is the sum of the resistances of the line segments.

- The resistance between the loads and the junction points are omitted.

The graph of the simplified network can be seen in Figure A.2. This network contains only 42 nodes and 41 edges.

Decomposition of the network

At first the decomposition algorithm is applied to the simplified graph model of the network. The result of the decomposition is 12 subnetworks. The subnetworks and the loads belonging to them can be seen in Table A.1 and in Figure A.2.

Table A.1: Subnetworks and the contained loads

Subnetwork	$S1$	$S2$	$S3$	$S4$	$S5$	$S6$	$S7$	$S8$	$S9$	$S10$	$S11$	$S12$
Loads	$L1$ $L3$	$L4$ $L5$	$L9$	$L14$	$L21$	$L20$ $L22$	$L34$	$L25$ $L30$	$L29$ $L31$	$L46$ $L48$ $L49$	$L52$ $L55$	$L51$ $L54$

It can be seen that there are 4 single load subnetworks ($S3, S4, S5, S7$) and 8 multiple load subnetworks ($S1, S2, S6, S8, S9, S10, S11, S12$) in the network.

A.2 Diagnosis of the illegal users

In this section the diagnostic algorithm is illustrated on the IEEE test network. The measured values are generated by simulation of the network in MATLAB. To simulate the presence of illegal users, the current of some loads are increased with respect to their nominal values. Measurement errors with zero mean Gaussian distribution are added to the simulated values to model the real measurements. The current and voltage measurement error limits are set to $\varepsilon_I = \varepsilon_U = 0.2\%$ of their nominal values. The uncertainty of the resistances is 2% of their nominal values.

For performing the diagnosis, let us consider the case, when the measured currents ($\tilde{I}$) and voltages ($\tilde{U}$) of the feeder and the loads are known and they are collected are in Table A.2. The nominal voltages (U) of the loads are computed using the measured currents of the loads.

Illegal load detection - 1 The sum of the measured currents of the loads is 30.217 A. The measurement error limit is $0.002(35.565 + 30.217) = 0.132$ A. The difference between the current of the transformer and the loads is 5.348 A, therefore illegal loads are detected in the network.

Then the difference between the measured and the nominal voltages are computed. The voltage differences of the loads can be seen in Figure A.3 with circle markers. The voltage differences are greater (in absolute value) than the acceptable deviation caused by the resistance uncertainties in the model. Therefore the voltage differences are caused by the illegal loads.

Localization of the illegal loads - 1 The diagnosis starts with the searching for illegal loads in the multiple load subnetworks. The illegal currents are calculated at the loads of the given multiple load subnetwork. Because the illegal currents are calculated based on the voltage differences of subsequent loads, the threshold of the illegal currents are set to $\varepsilon_U(\tilde{U}_i - \tilde{U}_{i+1})$ between two loads. This means that only those currents are considered as the result of

Table A.2: Measured and nominal values of the loads at the start of the diagnosis.

Load ID	$F1$	$L1$	$L3$	$L4$	$L5$	$L9$	$L14$	$L20$	$L21$	$L22$	$L25$	$L29$	$L30$	$L31$	$L34$	$L46$	$L48$	$L49$	$L51$	$L52$	$L54$	$L55$
$\tilde{I}$ [A]	35.565	2.466	1.509	0.737	1.825	2.220	0.205	0.214	0.189	1.053	1.659	0.802	0.223	0.659	7.978	1.536	1.116	0.642	0.638	2.036	0.919	1.591
$\tilde{U}$ [V]	252.187	251.624	251.614	251.154	251.147	250.958	250.863	250.588	250.709	250.577	250.265	250.255	250.261	250.257	250.154	250.334	250.324	250.319	250.377	250.265	250.367	250.254
U [V]	252.187	251.865	251.853	251.472	251.464	251.308	251.268	250.995	251.104	250.985	250.671	250.658	250.669	250.659	250.562	250.857	250.852	250.846	250.838	250.728	250.828	250.713

Table A.3: Measured and nominal values of the loads after finding one illegal load.

Load ID	$F1$	$L1$	$L3$	$L4$	$L5$	$L9$	$L14$	$L20$	$L21$	$L22$	$L25$	$L29$	$L30$	$L31$	$L34$	$L46$	$L48$	$L49$	$L51$	$L52$	$L54$	$L55$
$\tilde{I}$ [A]	35.565	2.466	1.509	0.737	1.825	2.220	0.205	0.214	0.189	1.053	1.659	0.802	0.223	0.659	7.978	1.536	3.772	0.642	0.638	2.036	0.919	1.591
$\tilde{U}$ [V]	252.187	251.624	251.614	251.154	251.147	250.958	250.863	250.588	250.709	250.577	250.265	250.255	250.261	250.257	250.154	250.334	250.324	250.319	250.377	250.265	250.367	250.254
U [V]	252.187	251.839	251.827	251.406	251.398	251.226	251.176	250.876	250.991	250.865	250.551	250.538	250.550	250.545	250.443	250.647	250.635	250.630	250.675	250.565	250.665	250.550

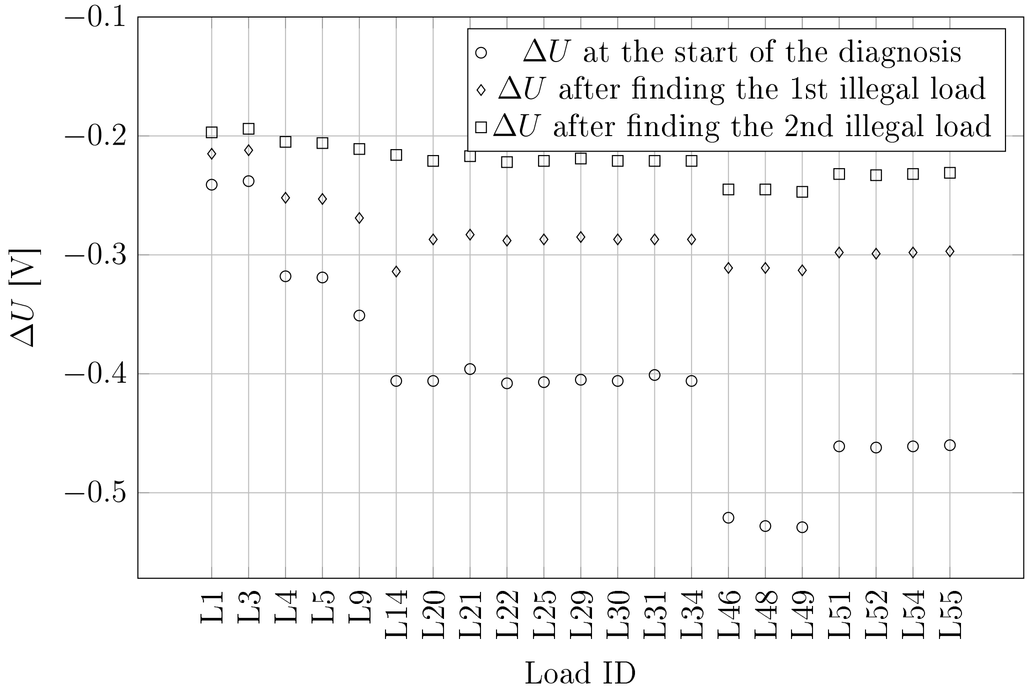


Figure A.3: The difference between the measured and the nominal voltages of the loads during the diagnosis.

illegal consumption that are greater than the given threshold. The result of this part of the diagnosis is that there is an illegal load at $L48$ and the illegal current is 2.656 A. The current of $L48$ is updated with the this illegal current so the current of $L48$ is 3.772 A. After that the network is simulated assuming the updated currents of the loads. The new nominal voltages can be seen in Table A.3.

Illegal load detection - 2 The detection criterion is still true: $35.565 - 32.873 > 0.002(35.565 + 32.873)$ therefore there are more illegal loads in the single load subnetworks. The difference between the measured and the new nominal voltages can be seen in Figure A.3 with diamond markers.

Localization of the illegal loads - 2 To find the illegal load(s) in the single load subnetworks, the voltage difference of the load is compared to the voltage difference of its two nearest neighbours. The nearest neighbours are determined using the length of the path (sum of the edge weights) between two nodes. If the voltage difference of the single load is smaller than its two neighbours then it is an illegal load. To take into account the effect of voltage measurement error a threshold is determined, too. Here the threshold is set to $\varepsilon_U(\tilde{U}_i - \tilde{U}_{i+1})$, because during the localization the voltage differences of neighbouring nodes are compared. The single loads and their neighbours are:

- $L9$: $L4$, $L14$
- $L14$: $L4$, $L20$

- $L21$: $L14$, $L20$
- $L34$: $L25$, $L20$

The result of this part of the diagnostic algorithm is an illegal load at $L14$. The estimated magnitude of the illegal current $0.068V/0.035\Omega = 1.9A$.

After correcting the current of $L14$ and simulating the network with the updated currents we get new nominal voltages. The difference between the measured and the nominal voltages can be seen in Figure A.3 with square markers. However the detection criterion based on comparing the currents is still true, the diagnostic algorithm cannot find any new illegal loads. In case of the multiple load subnetworks the computed illegal currents do not exceed the error threshold limit. In case of the single load subnetworks there is no local minimum between the voltage difference of the single loads and their neighbours. The missing current may come from two sources. It may happen that there are still illegal loads in the network but their current is too small. Besides that the computed illegal currents of $L48$ and $L14$ are not the accurate values but the approximations of the real values because of the measurement errors. Therefore the missing current may come from the approximation error too.

In conclusion two illegal loads are localized in the network. One of them is in a multiple load subnetwork and the other is in a single load subnetwork. The magnitude of the illegal currents are also estimated.

B Case study of colored Petri net based diagnosis of process systems

The system consists of three tanks in serial connection. The tanks are connected to each other with valves and pipes. Each tank has an input valve, an output valve and a level sensor. The technological system can be seen in Figure B.1 The variables belonging to each unit are listed below.

- TA: VA, VB, LA
- TB: VB, VC, LB
- TC: VC, VD, LC

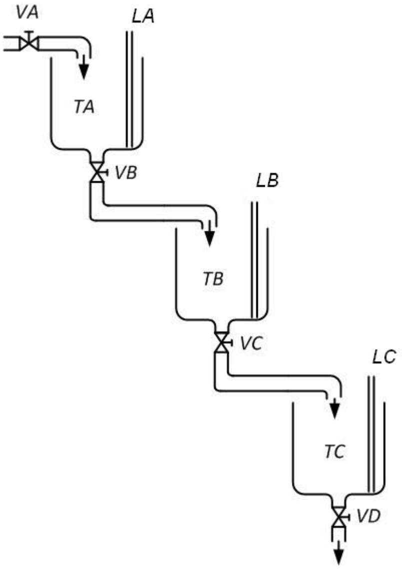


Figure B.1: The technological example

The qualitative range spaces used for the valves and the sensors are the following:

- valves: $Q_V = \{op, cl\}$ with the meaning of op = 'open' and cl = 'close';
- level sensors: $Q_L = \{e^-, 0, L, N, H, e^+\}$, where the meaning of $0, L, N, H$ are zero, low, normal, high and e^-, e^+ refer to the outlier values (below zero or above high).

The possible faults that can occur in each tank are the following:

- positive bias error of the level sensor;
- negative bias error of the level sensor;
- leak of the tank;
- the output valve only half opened when opened;

- the output valve stays closed when opened.

The examined technological process is the filling of the three tanks in a row. Initially all valves except VA are closed and the tanks are empty. The process starts with the filling of the first tank TA. After two time steps the water in TA reaches the normal level and the output valve VB is opened. Then the second tank TB starts filling up similarly. At time step 5 the output valve VC of the second tank is opened and the liquid flows into the third tank. Then the third tank is filled and finally its output valve VD is opened.

The nominal trace of the process can be seen in Table B.1 where each row refers to an event. It can be seen that the start and finish times of the units are the following:

- TA: start:1, finish: 3;
- TB: start:3, finish: 5;
- TC: start:5, finish: 7.

The nominal subtraces of the units after the decomposition are the following:

- TA: $[(1, op, cl, 0), (2, op, cl, L), (3, op, op, N)]$;
- TB: $[(1, op, cl, 0), (2, op, cl, L), (3, op, op, N)]$;
- TC: $[(1, op, cl, 0), (2, op, cl, L), (3, op, op, N)]$.

Note, that the start time of each unit is shifted back to 1 to get the subtraces.

The decomposition of the trace (with the original time instances) can be also seen in Table B.1 where the subtraces of TA, TB and TC are framed with solid, dashed and dotted lines respectively.

Table B.1: The full nominal trace of the system and its decomposition.

	Input variables				Output variables		
time	VA	VB	VC	VD	LA	LB	LC
1	*op*	*cl*	*cl*	*cl*	0	0	0
2	*op*	*cl*	*cl*	*cl*	L	0	0
3	*op*	*op*	*cl*	*cl*	N	0	0
4	*op*	*op*	*cl*	*cl*	N	L	0
5	*op*	*op*	*op*	*cl*	N	N	0
6	*op*	*op*	*op*	*cl*	N	N	L
7	*op*	*op*	*op*	*op*	N	N	N

Let us assume that the measured trace that can be seen in Table B.2 is observed during the operation of the system. The measured trace can be decomposed similarly to the nominal one:

- TA: $[(1, op, cl, L), (2, op, cl, N), (3, op, op, H)]$,
- TB: $[(1, op, cl, 0), (2, op, cl, 0), (3, op, op, 0)]$,

Table B.2: The full measured trace of the system and its decomposition.

time	Input variables				Output variables		
	VA	VB	VC	VD	LA	LB	LC
1	*op*	*cl*	*cl*	*cl*	*L*	0	0
2	*op*	*cl*	*cl*	*cl*	*N*	0	0
3	*op*	*op*	*cl*	*cl*	*H*	0	0
4	*op*	*op*	*cl*	*cl*	*H*	0	0
5	*op*	*op*	*op*	*cl*	*H*	0	0
6	*op*	*op*	*op*	*cl*	*H*	0	0
7	*op*	*op*	*op*	*op*	*H*	0	0

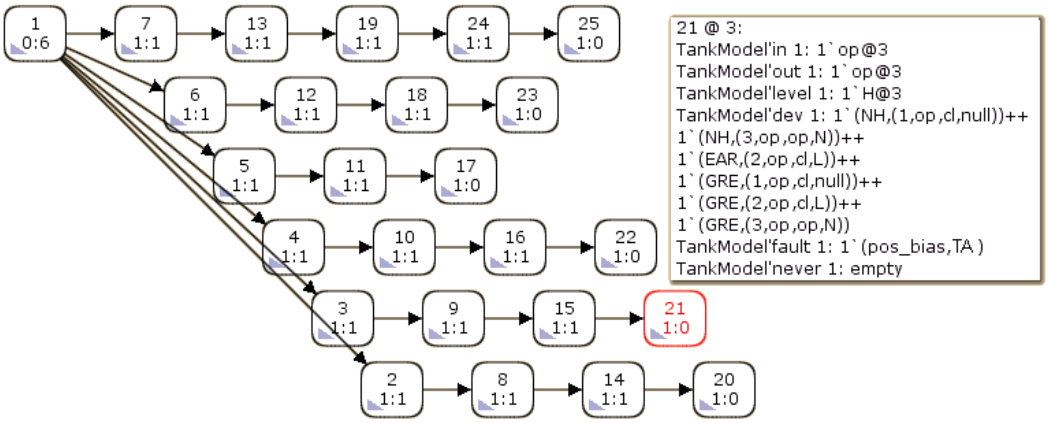
- TC: $[(1, op, cl, 0), (2, op, cl, 0), (3, op, op, 0)]$.

Now the diagnosis of the three tanks can be performed. Let us assume that we apply the diagnostic algorithm to the system at time 7 when all three tanks have finished their operation.

The diagnostic algorithm starts with the first tank (TA). The deviations between the nominal and the measured subtrace of this unit are the following:

$$EAR(2, opn, cl, L), GRE(1, opn, cl, null), GRE(2, opn, cl, L),$$
$$GRE(3, opn, opn, N), NH(1, opn, cl, null), NH(3, opn, opn, N)$$

The occurrence graph of the first tank can be seen in Figure B.2. The deviations are searched on the nodes of the occurrence graph and the node No. 21 is found with the same deviation list. The marking of this node is displayed in Figure B.2 too. It can be seen that the place $fault$ has a token with color $1`(pos\_bias, TA)$ therefore the positive bias error of the level sensor is identified in this unit.



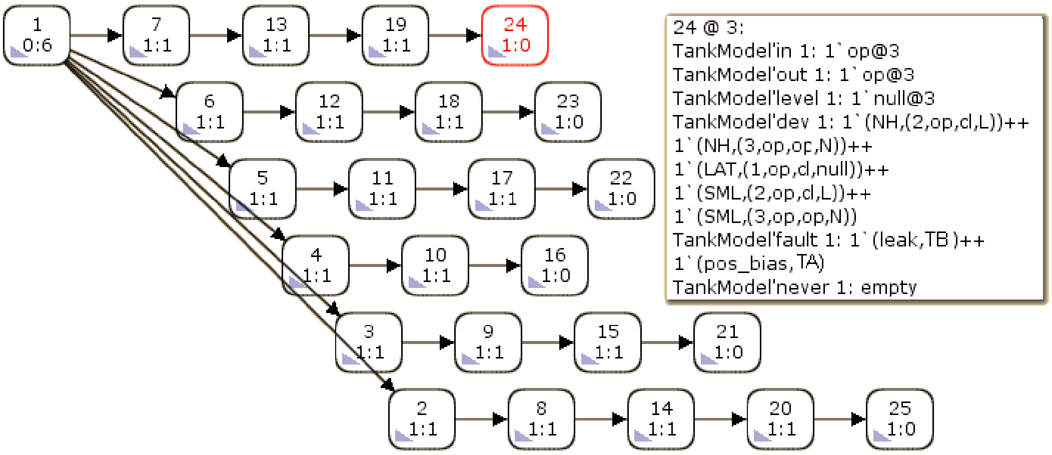
```
21 @ 3:
TankModel'in 1: 1`op@3
TankModel'out 1: 1`op@3
TankModel'level 1: 1`H@3
TankModel'dev 1: 1`(NH,(1,op,cl,null))++
1`(NH,(3,op,op,N))++
1`(EAR,(2,op,cl,L))++
1`(GRE,(1,op,cl,null))++
1`(GRE,(2,op,cl,L))++
1`(GRE,(3,op,op,N))
TankModel'fault 1: 1`(pos_bias,TA )
TankModel'never 1: empty
```

Figure B.2: Occurrence graph of the first tank

Then the diagnosis is continued with the second tank (TB). The deviations between the nominal and measured subtraces of TB are the following:

$$LAT(1, opn, cl, null), SML(2, opn, cl, L), SML(3, opn, opn, N),$$
$$NH(2, opn, cl, L), NH(3, opn, opn, N)$$

The positive bias error diagnosed in TA is added to place $fault$ in the CPN model of TB before the generation of the occurrence graph. The generated occurrence graph can be seen in Figure B.3. In this graph the node No. 24 has the same deviations list. The marking of place $fault$ at this node is $1`(pos\_bias, TA) + +1`(leak, TB)$ therefore a leakage in the second tank has occurred.



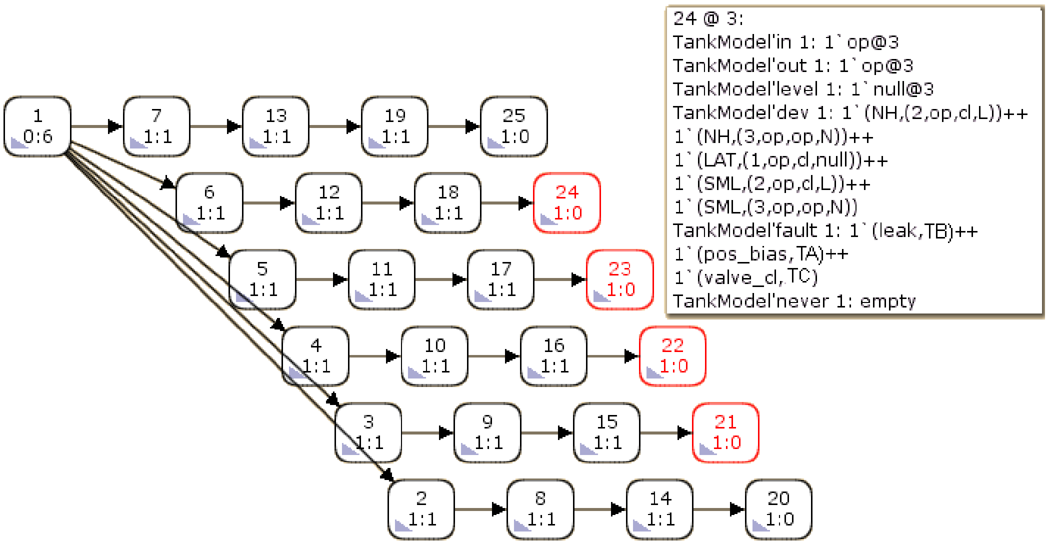
```
24 @ 3:
TankModel'in 1: 1`op@3
TankModel'out 1: 1`op@3
TankModel'level 1: 1`null@3
TankModel'dev 1: 1`(NH,(2,op,cl,L))++
1`(NH,(3,op,op,N))++
1`(LAT,(1,op,cl,null))++
1`(SML,(2,op,cl,L))++
1`(SML,(3,op,op,N))
TankModel'fault 1: 1`(leak,TB)++
1`(pos_bias,TA)
TankModel'never 1: empty
```

Figure B.3: Occurrence graph of the second tank

The last unit to be diagnosed is TC. The deviations at this tank are the following:

$$LAT(1, opn, cl, null), SML(2, opn, cl, L), SML(3, opn, opn, N),$$
$$NH(2, opn, cl, L), NH(3, opn, opn, N)$$

In this case the previously occurred faults $1`(pos\_bias, TA)++1`(leak, TB)$ are added as initial tokens to the place $fault$ in the CPN model of TC. Then the occurrence graph is generated with these initial conditions. The occurrence graph can be seen in Figure B.4. After searching the deviations on the graph four nodes have found with the given deviations: nodes No. 21, 22, 23 and 24. The corresponding faults are $(norm, TC)$, $(leak, TC)$, $(valve\_half, TC)$, $(valve\_cl, TC)$ which means that only a set of possible operation modes can be diagnosed. The actual operational mode cannot be clearly determined, the four possible operation modes are normal operation, leakage and two kinds of output valve errors.



```
24 @ 3:
TankModel'in 1: 1`op@3
TankModel'out 1: 1`op@3
TankModel'level 1: 1`null@3
TankModel'dev 1: 1`(NH,(2,op,cl,L))++
1`(NH,(3,op,op,N))++
1`(LAT,(1,op,cl,null))++
1`(SML,(2,op,cl,L))++
1`(SML,(3,op,op,N))
TankModel'fault 1: 1`(leak,TB)++
1`(pos_bias,TA)++
1`(valve_cl,TC)
TankModel'never 1: empty
```

Figure B.4: Occurrence graph of the third tank

In conclusion the following faults are diagnosed in the system:

- TA: positive bias sensor error;
- TB: leak;
- TC: operational mode is not clear: normal operation, leak, half opened output valve or closed output valve are all possible.

Bibliography

[1] H. Bourlès and G. K. Kwan, *Linear systems.* John Wiley & Sons, 2013.

[2] W. H. Kersting, *Distribution system modeling and analysis.* CRC press, 2006.

[3] C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems.* Springer Science & Business Media, 2009.

[4] H. K. Khalil and J. W. Grizzle, *Nonlinear systems.* Prentice hall Upper Saddle River, NJ, 2002, vol. 3.

[5] A. Avižienis, J.-C. Laprie, and B. Randell, "Dependability and its threats: A taxonomy", in *Building the Information Society*, Springer, 2004, pp. 91–120.

[6] R. J. Patton, P. M. Frank, and R. N. Clark, *Issues of fault diagnosis for dynamic systems.* Springer Science & Business Media, 2013.

[7] R. Isermann, *Fault-diagnosis systems: an introduction from fault detection to fault tolerance.* Springer Science & Business Media, 2006.

[8] M. Blanke, M. Kinnaert, and J. Lunze, *Diagnosis and fault-tolerant control.* Springer-Verlag, 2006.

[9] A. Pouliezos and G. Stavrakakis, "Parameter estimation methods for fault monitoring", in *Real Time Fault Monitoring of Industrial Processes*, Springer, 1994, pp. 179–255.

[10] O. Moseler andR. Isermann, "Application of model-based fault detection to a brushless DC motor", *IEEE Transactions on industrial electronics*, 47th vol., 5th no., pp. 1015–1020, 2000.

[11] E. M. Cimpoesu, B. D. Ciubotaru, and D. Stefanoiu, "Fault detection and diagnosis using parameter estimation with recursive least squares", in *2013 19th International Conference on Control Systems and Computer Science*, IEEE, 2013, pp. 18–23.

[12] Y. G. Li and T. Korakiantis, "Nonlinear weighted-least-squares estimation approach for gas-turbine diagnostic applications", *Journal of Propulsion and Power*, 27th vol., 2nd no., pp. 337–345, 2011.

[13] S. McIlraith, G. Biswas, D. Clancy, and V. Gupta, "Hybrid systems diagnosis", in *International Workshop on Hybrid Systems: Computation and Control*, Springer, 2000, pp. 282–295.

[14] S. Narasimhan and G. Biswas, "Model-based diagnosis of hybrid systems", *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and humans*, 37th vol., 3rd no., pp. 348–361, 2007.

[15] M. J. Rothenberger, D. J. Docimo, M. Ghanaatpishe, and H. K. Fathy, "Genetic optimization and experimental validation of a test cycle that maximizes parameter identifiability for a Li-ion equivalent-circuit battery model", *Journal of Energy Storage*, 4th vol., pp. 156–166, 2015.

[16] S. Mendoza, M. Rothenberger, J. Liu, and H. K. Fathy, "Maximizing parameter identifiability of a combined thermal and electrochemical battery model via periodic current input optimization", *IFAC-PapersOnLine*, 50th vol., 1st no., pp. 7314–7320, 2017.

[17] C. Kitsos, *Optimal Experimental Design for Non-Linear Models: Theory and Applications*, SpringerBriefs in Statistics ser. Springer Berlin Heidelberg, 2014, ISBN: 9783642452871.

[18] L. Wang, Z. Zhang, C. Huang, and K. L. Tsui, "A GPU-accelerated parallel Jaya algorithm for efficiently estimating Li-ion battery model parameters", *Applied Soft Computing*, 65th vol., pp. 12–20, 2018.

[19] M. Mathew, S. Janhunen, M. Rashid, F. Long, and M. Fowler, "Comparative analysis of lithium-ion battery resistance estimation techniques for battery management systems", *Energies*, 11th vol., 6th no., p. 1490, 2018.

[20] E. Chow and A. Willsky, "Analytical redundancy and the design of robust failure detection systems", *IEEE Transactions on automatic control*, 29th vol., 7th no., pp. 603–614, 1984.

[21] J. Gertler, "Analytical redundancy methods in fault detection and isolation-survey and synthesis", *IFAC Proceedings Volumes*, 24th vol., 6th no., pp. 9–21, 1991.

[22] A. Pouliezos and G. Stavrakakis, "Analytical redundancy methods", in *Real Time Fault Monitoring of Industrial Processes*, Springer, 1994, pp. 93–178.

[23] J. Gertler, "Structured residuals for fault isolation, disturbance decoupling and modelling error robustness", *IFAC Proceedings Volumes*, 25th vol., 4th no., pp. 15–23, 1992.

[24] D. Theilliol, H. Noura, and J.-C. Ponsart, "Fault diagnosis and accommodation of a three-tank system based on analytical redundancy", *ISA transactions*, 41st vol., 3rd no., pp. 365–382, 2002.

[25] R. Dorr, F. Kratz, J. Ragot, F. Loisy, and J.-L. Germain, "Detection, isolation, and identification of sensor faults in nuclear power plants", *IEEE Transactions on Control Systems Technology*, 5th vol., 1st no., pp. 42–60, 1997.

[26] W. C. Merrill, "Sensor failure detection for jet engines using analytical redundancy", *Journal of Guidance, Control, and Dynamics*, 8th vol., 6th no., pp. 673–682, 1985.

[27] R. Wang, Y. Cheng, and M. Xu, "Analytical redundancy based fault diagnosis scheme for satellite attitude control systems", *Journal of the Franklin Institute*, 352nd vol., 5th no., pp. 1906–1931, 2015.

[28] R. J. Patton, “Fault detection and diagnosis in aerospace systems using analytical redundancy”, *Computing & Control Engineering Journal*, 2nd vol., 3rd no., pp. 127–136, 1991.

[29] E. Sobhani-Tehrani and K. Khorasani, *Fault diagnosis of nonlinear systems using a hybrid approach.* Springer Science & Business Media, 2009, vol. 383.

[30] J. Zaytoon and S. Lafortune, “Overview of fault diagnosis methods for discrete event systems”, *Annual Reviews in Control*, 37th vol., 2nd no., pp. 308–320, 2013.

[31] M. Sampath,R. Sengupta,S. Lafortune,K. Sinnamohideen, andD. Teneketzis, “Diagnosability of discrete-event systems”, *Automatic Control, IEEE Transactions on*, 40th vol., 9th no., pp. 1555–1575, 1995.

[32] S. H. Zad, R. H. Kwong, and W. M. Wonham, “Fault diagnosis in discrete-event systems: Framework and model reduction”, *IEEE Transactions on Automatic Control*, 48th vol., 7th no., pp. 1199–1212, 2003.

[33] J. Lunze and P. Supavatanakul, “Diagnosis of discrete–event system described by timed automata”, *IFAC Proceedings Volumes*, 35th vol., 1st no., pp. 77–82, 2002.

[34] D. Förstner and J. Lunze, “Discrete-event models of quantized systems for diagnosis”, *International Journal of Control*, 74th vol., 7th no., pp. 690–700, 2001.

[35] J. Prock, “A new technique for fault detection using Petri nets”, *Automatica*, 27th vol., 2nd no., pp. 239–245, 1991.

[36] A. Ramirez-Trevino, E. Ruiz-Beltran, I. Rivera-Rangel, and E. Lopez-Mellado, “Online fault diagnosis of discrete event systems. A Petri net-based approach”, *IEEE Transactions on Automation Science and Engineering*, 4th vol., 1st no., pp. 31–39, Jan. 2007.

[37] M. Alcaraz-Mejia,E. Lopez-Mellado,A. Ramírez-Treviño, andI. Rivera-Rangel, “Petri net based fault diagnosis of discrete event systems”, in *SMC’03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme-System Security and Assurance (Cat. No. 03CH37483)*, IEEE, vol. 5, 2003, pp. 4730–4735.

[38] I. Fliss andM. Tagina, “Multiple fault diagnosis of discrete event systems using Petri nets”, in *2011 International Conference on Communications, Computing and Control Applications (CCCA)*, IEEE, 2011, pp. 1–6.

[39] M. Dotoli, M. P. Fanti, A. M. Mangini, and W. Ukovich, “On-line fault detection in discrete event systems by Petri nets and integer linear programming”, *Automatica*, 45th vol., 11th no., pp. 2665–2672, 2009.

[40] D. Lefebvre and C. Delherm, “Diagnosis of DES with Petri net models”, *IEEE Transactions on Automation Science and Engineering*, 4th vol., 1st no., pp. 114–118, 2007.

[41] A. Fanni and A. Giua, “Discrete event representation of qualitative models using Petri nets”, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 28th vol., 6th no., pp. 770–780, 1998.

[42] M. Gerzson, Z. Csaki, and K. Hangos, “Qualitative model based verification of operating procedures by high level Petri nets”, *Computers & chemical engineering*, 18th vol., S565–S569, 1994.

[43] A. Tóth, E. Németh, and K. M. Hangos, “Coloured Petri net diagnosers for lumped process systems”, in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, Springer, 2010, pp. 389–398.

[44] S. S. Madani, E. Schaltz, and S. K. Kær, “A review of different electric equivalent circuit models and parameter identification methods of lithium-ion batteries”, *Ecs Transactions*, 87th vol., 1st no., pp. 23–37, 2018.

[45] S. Panchal, M. Mathew, R. Fraser, and M. Fowler, “Electrochemical thermal modeling and experimental measurements of 18650 cylindrical lithium-ion battery during discharge cycle for an EV”, *Applied Thermal Engineering*, 135th vol., pp. 123–132, 2018.

[46] **A. I. Pózna**, A. Magyar, and K. M. Hangos, “Model identification and parameter estimation of lithium ion batteries for diagnostic purposes”, in *2017 International Symposium on Power Electronics (Ee)*, IEEE, 2017, pp. 1–6.

[47] G. L. Plett, *Battery management systems, Volume I: Battery modeling.* Artech House, 2015, vol. 1.

[48] D. Andre, C. Appel, T. Soczka-Guth, and D. U. Sauer, “Advanced mathematical methods of SOC and SOH estimation for lithium-ion batteries”, *Journal of Power Sources*, 224th vol., pp. 20–27, 2013.

[49] A. Eddahech, O. Briat, and J. M. Vinassa, “Real-time SOC and SOH estimation for EV Li-ion cell using online parameters identification”, in *Proc. IEEE Energy Conversion Congress and Exposition (ECCE)*, Sep. 2012, pp. 4501–4505.

[50] X. Hu, S. Li, H. Peng, and F. Sun, “Robustness analysis of State-of-Charge estimation methods for two types of Li-ion batteries”, *Journal of Power Sources*, 217th vol., pp. 209–219, 2012.

[51] Y. Hu and S. Yurkovich, “Battery state of charge estimation in automotive applications using LPV techniques”, in *American Control Conference (ACC), 2010*, IEEE, 2010, pp. 5043–5049.

[52] Y. Xing, W. He, M. Pecht, and K. L. Tsui, “State of charge estimation of lithium-ion batteries using the open-circuit voltage at various ambient temperatures”, *Applied Energy*, 113th vol., pp. 106–115, 2014.

[53] J. Groot, *State-of-health estimation of Li-ion batteries: Cycle life test methods*, 2012.

[54] J. Zhang and J. Lee, “A review on prognostics and health monitoring of Li-ion battery”, *Journal of Power Sources*, 196th vol., 15th no., pp. 6007–6014, 2011.

[55] O. Tremblay, L.-A. Dessaint, and A.-I. Dekkiche, “A generic battery model for the dynamic simulation of hybrid electric vehicles”, in *Vehicle Power and Propulsion Conference, 2007. VPPC 2007. IEEE*, 2007, pp. 284–289.

[56] **A. I. Pózna**, K. M. Hangos, and A. Magyar, “Design of experiments for battery aging estimation”, *IFAC-PapersOnLine*, 51st vol., 28th no., pp. 386–391, 2018.

[57] *Simscape version 4.3 (R2017a)*, The Mathworks, Inc., Natick, Massachusetts, 2017.

[58] O. Tremblay and L.-A. Dessaint, “Experimental validation of a battery dynamic model for EV applications”, *World Electric Vehicle Journal*, 3rd vol., 1st no., pp. 1–10, 2009.

[59] *Simulink version: 9.3*, The Mathworks, Inc., Natick, Massachusetts, 2019.

[60] L. Ljung, *System identification: Theory for the user (2nd edition)*. New Jersey: Prentice Hall Inc., 1999.

[61] L. Saw, K. Somasundaram, Y. Ye, and A. Tay, “Electro-thermal analysis of lithium iron phosphate battery for electric vehicles”, *Journal of Power Sources*, 249th vol., pp. 231–238, 2014.

[62] R. H. Byrd, R. B. Schnabel, and G. A. Shultz, “A trust region algorithm for nonlinearly constrained optimization”, *SIAM Journal on Numerical Analysis*, 24th vol., 5th no., pp. 1152–1170, 1987.

[63] K. Yang, J. J. An, and S. Chen, “Temperature characterization analysis of LiFePO 4/c power battery during charging and discharging”, *Journal of thermal analysis and calorimetry*, 99th vol., 2nd no., pp. 515–521, 2010.

[64] *MATLAB Optimization Toolbox version: 9.6.0,1047502 (R2019a)*, The Mathworks, Inc., Natick, Massachusetts, 2019.

[65] *MATLAB Curve Fitting Toolbox version: 9.6.0,1047502 (R2019a)*, The Mathworks, Inc., Natick, Massachusetts, 2019.

[66] T. Ahmad, H. Chen, J. Wang, and Y. Guo, “Review of various modeling techniques for the detection of electricity theft in smart grid environment”, *Renewable and Sustainable Energy Reviews*, 82nd vol., pp. 2916 –2933, 2018, ISSN: 1364-0321.

[67] J Nagi, A. Mohammad, K. Yap, S. Tiong, and S. Ahmed, “Non-technical loss analysis for detection of electricity theft using support vector machines”, in *Power and Energy Conference, 2008. PECon 2008. IEEE 2nd International*, IEEE, 2008, pp. 907–912.

[68] T. B. Smith, “Electricity theft: A comparative analysis”, *Energy Policy*, 32nd vol., 18th no., pp. 2067 –2076, 2004.

[69] G. M. Messinis and N. D. Hatziargyriou, “Review of non-technical loss detection methods”, *Electric Power Systems Research*, 158th vol., pp. 250 –266, 2018, ISSN: 0378-7796.

[70] J. L. Viegas, P. R. Esteves, R. Melício, V. Mendes, and S. M. Vieira, “Solutions for detection of non-technical losses in the electricity grid: A review”, *Renewable and Sustainable Energy Reviews*, 80th vol., pp. 1256 –1268, 2017, ISSN: 1364-0321.

[71] S.-C. Yip, K. Wong, W.-P. Hew, M.-T. Gan, R. C.-W. Phan, and S.-W. Tan, “Detection of energy theft and defective smart meters in smart grids using linear regression”, *International Journal of Electrical Power & Energy Systems*, 91st vol., pp. 230 –240, 2017, ISSN: 0142-0615.

[72] A. A. Ghasemi and M. Gitizadeh, “Detection of illegal consumers using pattern classification approach combined with Levenberg-Marquardt method in smart grid”, *International Journal of Electrical Power & Energy Systems*, 99th vol., pp. 363 –375, 2018, ISSN: 0142-0615.

[73] S. S. S. R. Depuru, L. Wang, V. Devabhaktuni, and R. C. Green, “High performance computing for detection of electricity theft”, *International Journal of Electrical Power & Energy Systems*, 47th vol., pp. 21 –30, 2013, ISSN: 0142-0615.

[74] A. Pasdar andS. Mirzakuchaki, “A solution to remote detecting of illegal electricity usage based on smart metering”, in *2007 2nd International Workshop on Soft Computing Applications*, 2007, pp. 163–167.

[75] S. A. Salinas and P. Li, “Privacy-preserving energy theft detection in microgrids: A state estimation approach”, *IEEE Transactions on Power Systems*, 31st vol., 2nd no., pp. 883–894, 2016.

[76] E. A. A. Neto and J. Coelho, “Probabilistic methodology for technical and non-technical losses estimation in distribution system”, *Electric Power Systems Research*, 97th vol., pp. 93–99, 2013.

[77] K. Dasgupta, M. Padmanaban, and J. Hazra, “Power theft localisation using voltage measurements from distribution feeder nodes”, *IET Generation, Transmission & Distribution*, 11th vol., 11th no., pp. 2831–2839, 2017.

[78] **A. I. Pózna**, A. Fodor, M. Gerzson, and K. M. Hangos, “Colored Petri net model of electrical networks for diagnostic purposes”, *IFAC-PapersOnLine*, 51st vol., 2nd no., pp. 260–265, 2018.

[79] C. Lara, D. Mallapragada, D. Papageorgiou, A. Venkatesh, and I. Grossmann, “Deterministic electric power infrastructure planning: Mixed-integer programming model and nested decomposition algorithm”, *Eur J Oper Res.*, 271st vol., pp. 1037–1054, 2018.

[80] J. Guo, G. Hug, and O. Tonguz, “Intelligent partitioning in distributed optimization of electric power systems”, *IEEE Trans Smart Grid*, 7th vol., pp. 1249–1258, 2016.

[81] S. Heo, S. Rangarajan, P. Daoutidis, and S. Jogwar, “Graph reduction of complex energy-integrated networks: Process systems applications”, *AIChE*, 60th vol., pp. 995–1012, 2014.

[82] S. Jogwar, S. Rangarajan, and P. Daoutidis, “Reduction of complex energy-integrated process networks using graph theory”, *Comput Chem Engng*, 79th vol., pp. 46–58, 2015.

[83] J. Bird, *Electrical circuit theory and technology*, 4th. Elsevier, 2010.

[84] F. C. Schweppe and J. Wildes, “Power system static-state estimation, Part I: Exact model”, *IEEE Transactions on Power Apparatus and systems*, 1st no., pp. 120–125, 1970.

[85] F. C. Schweppe and D. B. Rom, “Power system static-state estimation, Part II: Approximate model”, *IEEE Transactions on Power Apparatus and Systems*, 1st no., pp. 125–130, 1970.

[86] P. M. Frank, E. A. Garcia, and B. Kappen-Seliger, “Modelling for fault detection and isolation versus modelling for control”, *Mathematical and Computer Modelling of Dynamical Systems*, 7th vol., 1st no., pp. 1–46, 2001.

[87] G. Schullerus, P. Supavatanakul, V. Krebs, and J. Lunze, “Modelling and hierarchical diagnosis of timed discrete-event systems”, *Mathematical and Computer Modelling of Dynamical Systems*, 12th vol., 6th no., pp. 519–542, 2006.

[88] D. Matko, G. Geiger, and T. Werner, “Neural net versus classical models for the detection and localization of leaks in pipelines”, *Mathematical and Computer Modelling of Dynamical Systems*, 12th vol., 6th no., pp. 505–517, 2006.

[89] W. Borutzky, “Bond graph model-based system mode identification and mode-dependent fault thresholds for hybrid systems”, *Mathematical and Computer Modelling of Dynamical Systems*, 20th vol., 6th no., pp. 584–615, 2014.

[90] A. Kłos, *Mathematical Models of Electrical Network Systems: Theory and Applications-An Introduction.* Springer, 2017, vol. 412.

[91] M. Hultmark and A. J. Smits, “Temperature corrections for constant temperature and constant current hot-wire anemometers”, *Measurement Science and Technology*, 21st vol., 10th no., p. 105 404, 2010.

[92] V. Bartolozzi, L. Castiglione, A. Picciotto, and M. Galluzzo, “Qualitative models of equipment units and their use in automatic HAZOP analysis”, *Reliability Engineering & System Safety*, 70th vol., 1st no., pp. 49–57, 2000.

[93] R. Srinivasan and V. Venkatasubramanian, “Petri net-digraph models for automating HAZOP analysis of batch process plants”, *Computers & chemical engineering*, 20th vol., S719–S725, 1996.

[94] G. Jiroveanu and R. K. Boel, “A distributed approach for fault detection and diagnosis based on time Petri nets”, *Mathematics and Computers in Simulation*, 70th vol., 5th no., pp. 287–313, 2006.

[95] S. Genc and S. Lafortune, “Distributed diagnosis of place-bordered Petri nets”, *IEEE Transactions on Automation Science and Engineering*, 4th vol., 2nd no., pp. 206–219, Apr. 2007.

[96] Y. Pencolé, R. Pichard, and P. Fernbach, “Modular fault diagnosis in discrete-event systems with a CPN diagnoser”, *IFAC-PapersOnLine*, 48th vol., 21st no., pp. 470–475, 2015.

[97] F. Basile, P. Chiacchio, and G. D. Tommasi, “An efficient approach for online diagnosis of discrete event systems”, *IEEE Transactions on Automatic Control*, 54th vol., 4th no., pp. 748–759, Apr. 2009.

[98] M. P. Fanti, A. M. Mangini, and W. Ukovich, “Fault detection by labeled Petri nets and time constraints”, in *Dependable Control of Discrete Systems (DCDS), 2011 3rd International Workshop on*, Jun. 2011, pp. 168–173.

[99] D. Lefebvre and E. Aguayo-Lara, “Initial study for observers application to fault detection and isolation with continuous timed petri nets”, *IFAC-PapersOnLine*, 48th vol., 7th no., pp. 97–103, 2015.

[100] M. P. Cabasino, A. Giua, and C. Seatzu, “Fault detection for discrete event systems using Petri nets with unobservable transitions”, *Automatica*, 46th vol., 9th no., pp. 1531–1539, 2010.

[101] M. Bouali, P. Barger, and W. Schon, “Backward reachability of colored Petri nets for systems diagnosis”, *Reliability Engineering & System Safety*, 99th vol., pp. 1–14, 2012.

[102] T. Murata, “Petri nets: Properties, analysis and applications”, *Proceedings of the IEEE*, 77th vol., 4th no., pp. 541–580, 1989.

[103] K. Jensen, *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use.* Springer, 1996, vol. Volume 1.

[104] K. University of Aarhus, Denmark, CPN Group. (2016). CPNTools, [Online]. Available: `http://wiki.daimi.au.dk/cpntools/`.

[105] A. Tóth, K. Hangos, and A. Werner-Stark, “HAZID information based operational procedure diagnosis method”, in *12th International PhD Workshop on Systems and Control, Veszprém*, 2012, pp. 1–6.

[106] A. Tóth and K. M. Hangos, “A diagnostic method based on clustering qualitative event sequences”, *Computers & Chemical Engineering*, 95th vol., pp. 58–70, 2016.

[107] F. Krger and S. Merz, “Temporal logic and state systems (Texts in theoretical computer science. An EATCS series)”, 2008.

[108] S. Jiang andR. Kumar, “Diagnosis of repeated failures for discrete event systems with linear-time temporal-logic specifications”, *IEEE Transactions on Automation Science and Engineering*, 3rd vol., 1st no., pp. 47–59, 2006.

[109] K. Schneider, B. Mather, B. C. Pal, C.-W. Ten, G. Shirek, H. Zhu, J. Fuller, J. L. R. Pereira, L. Ochoa, L. De Araujo, *et al.*, “Analytic considerations and design basis for the IEEE distribution test feeders”, *IEEE Transactions on Power Systems*, 33rd vol., 3rd no., pp. 3181–3188, 2018.

[110] *Matlab version 9.0.0.341360 (R2016a)*, The Mathworks, Inc., Natick, Massachusetts, 2016.