Security Constrained Optimal Power Flow

Formulations, Challenges, and Methods

by

Jules Fleuren

Supervisor TU Delft:
Prof.dr.ir. C. Vuik

Supervisor TenneT:
S. Chipli

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
AC	Alternating Current
API	Application Programming Interface
DC	Direct Current
DSO	Distribution System Operator
ILP	Integer Linear Program
LP	Linear Program
MINLP	Mixed Integer Nonlinear Program
NDC	Non Dominated Contingency
NLP	Nonlinear Program
KCL	Kirchoff's Current Law
OPF	Optimal Power Flow
pu	per unit
RMS	Root Mean Square
SCOPF	Security Constrained Optimal Power Flow
TSO	Transmission System Operator

Symbols

Symbol	Definition	Unit
$\mathcal{E}$	set of all edges	-
i	instantaneous current	kA
I	current phasor	kA or pu
$\lvert I \rvert$	current magnitude	kA or pu
I_{max}	current amplitude	kA
$I^{\text{f}}_{k,l}$	current flowing out of node k into edge $\{k, l\}$	kA or pu
$I^{\text{t}}_{k,l}$	current flowing into node l out of edge $\{k, l\}$	kA or pu
j	imaginary unit	-
N	number of nodes	-
$\mathcal{N}$	set of all nodes	-
P	active power	MW or pu
P_i	active power injection at node i	MW or pu
Q	reactive power	Mvar or pu
Q_i	reactive power injection at node i	Mvar or pu
S	complex power	MVA or pu
t	time	s
t_{kl}	transformer tap ratio of edge $\{k, l\}$ at side of node k	-
T	system period	s
v	instantaneous voltage	kV
V	voltage phasor	kV or pu
V_i	voltage phasor at node i	kV or pu
$\lvert V \rvert$	normalized voltage magnitude	kV or pu
V_{max}	voltage amplitude	kV
Y	admittance matrix	Ω^{-1} or pu

Symbol	Definition	Unit
Y_{kl}^{es}	shunt admittance of edge $\{k,l\}$	Ω^{-1} or pu
Y_{i}^{ns}	shunt admittance at node i	Ω^{-1} or pu
Y_{kl}^{sr}	series admittance of edge $\{k,l\}$	Ω^{-1} or pu
Z	impedance	Ω
δ_i	voltage phase shift at node i	rad
θ_{kl}	transformer tap ratio phase shift	rad
τ_{kl}	transformer tap ratio magnitude	-
ϕ	voltage and current phase difference	rad
ψ_I	current phase shift	rad
ψ_V	voltage phase shift	rad
ω	System frequency	Hz

1 Introduction

TenneT is a Transmission System Operator (TSO), responsible for the transmission grid in the Netherlands and a large part of Germany. This responsibility includes maintaining, operating, and where necessary, expanding the transmission grid. With the energy transition from fossil fuels to renewable energy sources, which are mostly electric, the demand on the electricity grid increases. In addition to this, a transition from centrally generated energy, with a few big power plants, to distributed generation, with smaller scale solar and wind generation in many places, is taking place. Both of these transitions come with big challenges in the operation and expansion of the grid. To make efficient use of the existing assets and to ensure a stable and reliable operation of the grid, it is crucial to analyse the capabilities of the grid and optimize the control of the grid.

In this chapter, we sketch the position and role of TenneT in the electricity system and give some examples of challenges that TenneT faces in fulfilling these roles.

1.1. Position of TenneT

TenneT is the only TSO in the Netherlands. It is also one of the 4 TSO's operating the German transmission grid, but our focus will be on the Dutch grid. In figure 1.1 a map of the Dutch transmission grid can be seen. As a TSO, TenneT is responsible for the maintenance and operation of the high voltage transmission grid, that is, all parts of the grid that are 110 kV or higher.

Energy is traded on the electricity markets by suppliers and consumers of electricity, and this energy is then transported over the grid. TenneT does not participate in this market by buying or selling energy, but its responsibility is to transport this energy from suppliers to consumers. It can only intervene in the market if the stability of the grid cannot be ensured. These responsibilities and possibilities are determined by law, and TenneT is under supervision of the governmental agency *Autoriteit Consument en Markt*.

Parties that are directly connected to the high voltage transmission grid of TenneT are large power plants, or wind or solar parks with a generation of about 10 MW or higher, large industry with an energy consumption of about 10 MW or higher, and *Distribution System Operators* (DSO) that further distribute electricity to smaller scale industry and homes [30, sec. 2.1].

1.1.1. Ancillary Services

To ensure a stable and reliable transportation of energy, TenneT buys, builds and maintains its own assets, such as transmission lines, substations, and a variety of electrical components. In addition to this, it can however also make use of so-called *Ancillary Services*. These are services provided by parties connected to the grid, that TenneT uses to operate the grid in a stable way. Examples of ancillary services are: providing reserve generation capacity for active power balance, reactive power generation and scaling power generation up or down in case of redispatch. In section 1.2, we elaborate on two situations in which the stability of the grid is at risk and how TenneT makes use of ancillary services in these situations, to ensure the stability of the grid.

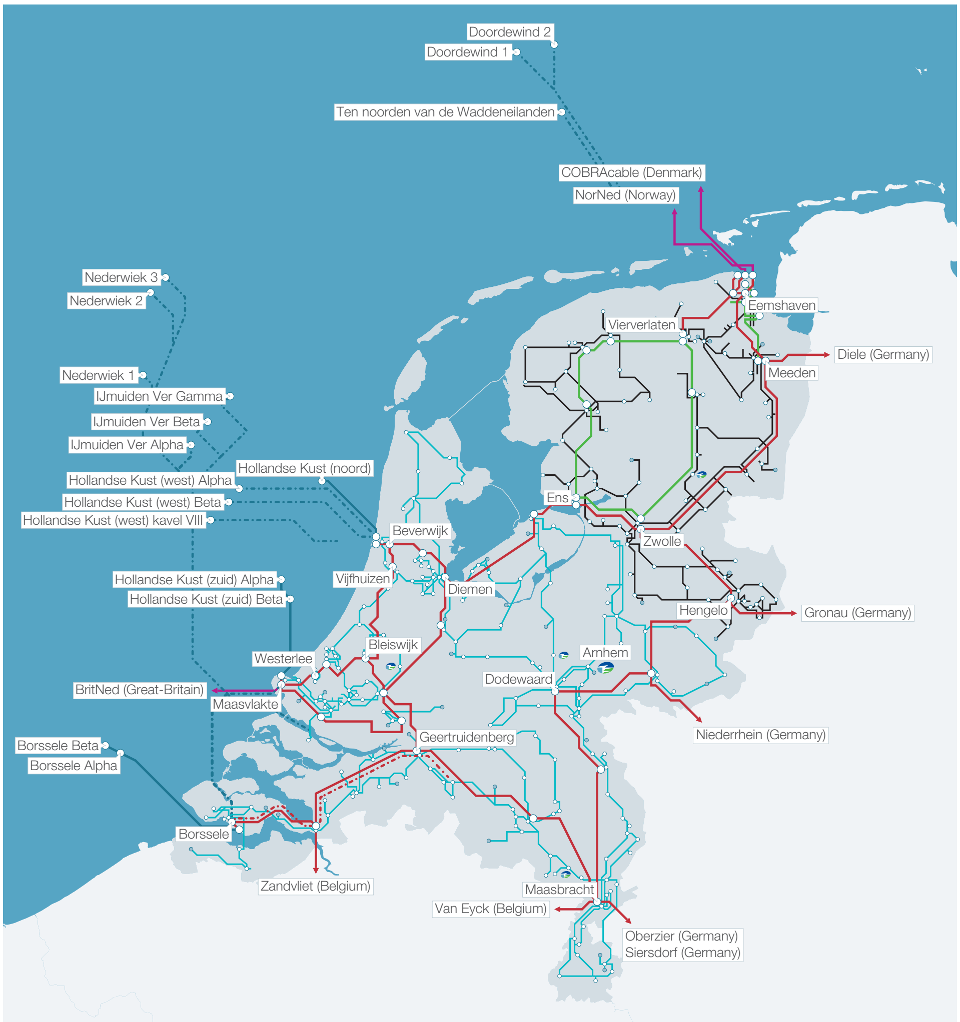

Figure 1.1: Map of the Dutch transmission grid, including connections to neighbouring countries and offshore wind farms. The dashed lines are planned connections. Reproduced from [3].

1.2. Operational Challenges

We introduce two challenges in operating the grid, that TenneT faces on a daily basis. Both challenges illustrate how TenneT has to actively and optimally control the grid, to ensure a stable and reliable transmission grid, at a low cost.

1.2.1. Grid Congestion

When the volume of power transported over the grid is high, and the locations of high supply and demand are not well distributed over the grid, *grid congestion* can occur. This means that the grid cannot transport all energy without overloading components in the grid. An example of a country where this happens frequently, is Germany, where high amounts of energy is generated by offshore wind parks in the north, and a lot of energy is consumed by heavy industry in the south. When the grid cannot safely transport all energy, a *redispatch* is performed. This means that generation is scaled down in parts of the grid with high generation, and instead energy generation located closer to the loads is scaled up. This ensures that all energy demand is still satisfied, while at the same time relieving some strain on the grid, because the energy needs to travel less far. Both the party scaling down their generation and the party scaling up their generation are financially compensated for this. These costs are paid by the TSO, which in turn raises its tariffs to cover the costs. This means that optimally utilizing the grid, and reducing the dependence on redispatch, can reduce costs for all electricity users. In practice, during redispatch, often fossil fuel powered energy is being scaled up, and renewable energy generation is being scaled down.

This means that reducing the dependence on redispatch could also reduce strain on the environment, by decreasing greenhouse gas emissions.

1.2.2. Reactive Power Balance

In contrast to active power, reactive power[1] is not traded on the energy market. Making sure supply and demand of reactive power are matched, is a task of TenneT. A lack of supply of reactive power can lead to voltage drops, and a surplus can lead to voltage rises. Both are undesirable. The task of maintaining this reactive power balance is called *reactive power dispatch*, or *voltage control*. Capacitive and reactive shunts can be activated to either supply or consume reactive power, or ancillary service providers can be asked to supply or consume reactive power. Some ancillary service providers can be asked to maintain a fixed voltage, and thus consume or supply reactive power as needed to maintain this voltage, others can be asked to supply or consume a specified amount of reactive power. Shunts can be activated without additional costs, but ancillary service providers are compensated financially for the amount of reactive power they provide or consume. Determining how shunts should be controlled, and how much reactive power should be requested from ancillary service providers, ensuring stable voltage levels at minimal cost, is a challenging problem.

[1] For an introduction to active and reactive power, see sec. 2.1.2.

2

Modelling the Transmission Grid

To be able mathematically analyse the grid, and to predict its behaviour, we introduce a mathematical model of a transmission system. First we introduce some concepts from the theory of Alternating Current (AC) circuits, then we describe how we can model a transmission grid as graph, and finally, we introduce *contingencies*, an important concept in the operation and modelling of transmission grids.

2.1. AC Circuit Fundamentals

Since the transmission of power via the power grid is mostly done with alternating current, we give an introduction to the behaviour of AC-circuits and how they can be modelled. We mostly follow the reasoning and notation of [24], [33] and [13].

2.1.1. Voltage and Current

In an AC-circuit, voltage and current vary in time following a sinusoidal function. In steady state circuit analysis, we assume that the frequency of these functions is fixed. Consider any part of a transmission network, that is connected to the rest of the network at two terminals. We can consider the voltage across the two terminals and the current flowing through the component as a function of time. The voltage is as follows:

$$v(t) = V_{\max}\cos(\omega t + \psi_V) \tag{2.1}$$

and the current is as follows:

$$i(t) = I_{\max}\cos(\omega t + \psi_I). \tag{2.2}$$

Here,

$$\begin{aligned} V_{\max} &= \text{voltage amplitude, kV} \\ I_{\max} &= \text{current amplitude, kA} \\ \omega &= \text{angular frequency, Hz} \\ t &= \text{time, s} \\ \psi_V &= \text{voltage phase shift, rad} \\ \psi_I &= \text{current phase shift, rad.} \end{aligned}$$

The time is measured to some reference time, where $t = 0$. In Europe, the electricity grid operates at a frequency of 50 Hz, this means that the angular frequency is $\omega = 2\pi \cdot 50\,\text{Hz}$. We define $\phi = \psi_V - \psi_I$ as the phase difference between the voltage and current. If ϕ is positive, we say that the voltage is leading the current and if ϕ is negative, the voltage is lagging the current.

In equations (2.1) and (2.2), $I_{\max}$ and $V_{\max}$ denote the amplitude of the current and voltage, respectively. However, in the context of AC electronics, it is customary to work with the Root Mean Square (RMS) of power and current [33, p. 7]. They can be calculated as follows:

$$|V| = \sqrt{\frac{1}{T}\int_0^T v(t)^2 dt} \tag{2.3}$$

$$|I| = \sqrt{\frac{1}{T}\int_0^T i(t)^2 dt} \tag{2.4}$$

here $T = 2\pi/\omega$ is the period of the sine waves. Substituting (2.1) and (2.2) in these formulas gives the relations $V_{max} = \sqrt{2}|V|$ and $I_{max} = \sqrt{2}|I|$.

Phasor Notation

If we rewrite equations (2.1) and (2.2) using Euler's identity, we get the following:

$$\begin{aligned} v(t) &= \sqrt{2}|V|\cos(\omega t + \psi_V) \\ &= \sqrt{2}\,\mathrm{Re}\left\{|V|e^{j\psi_V}e^{j\omega t}\right\} \\ &= \sqrt{2}\,\mathrm{Re}\left\{Ve^{j\omega t}\right\} \end{aligned} \qquad \begin{aligned} i(t) &= \sqrt{2}|I|\cos(\omega t + \psi_I) \\ &= \sqrt{2}\,\mathrm{Re}\left\{|I|e^{j\psi_I}e^{j\omega t}\right\} \\ &= \sqrt{2}\,\mathrm{Re}\left\{Ie^{j\omega t}\right\}. \end{aligned}$$

Here $V = |V|e^{\psi_V}$ and $I = |I|e^{\psi_I}$. Note that these equations can be used both ways, so we obtain the following one-to-one relationship between the complex numbers, and steady state voltage functions:

$$V = |V|e^{j\psi_V} \qquad \leftrightarrow \qquad v(t) = \sqrt{2}|V|\cos(\omega t + \psi_V) \tag{2.5}$$

and, similarly, between the complex numbers and steady state current functions:

$$I = |I|e^{j\psi_I} \qquad \leftrightarrow \qquad i(t) = \sqrt{2}|I|\cos(\omega t + \psi_I). \tag{2.6}$$

We call V and I the *voltage* and *current phasors*, respectively. In the rest of this report, we will denote the steady state voltage and current functions at some point in the grid by its voltage and current phasors. For more information, and a derivation of calculation rules for phasors, we refer to [33, sec. 1.4].

2.1.2. Power

The instantaneous power $p(t)$ consumed by a load through which an alternating current $i(t)$ flows across a voltage of $v(t)$, is calculated as follows:

$$\begin{aligned} p(t) &= v(t)i(t) && (2.7) \\ &= \sqrt{2}|V|\cos(\omega t + \psi_V)\sqrt{2}|I|\cos(\omega t + \psi_V - \phi) && (2.8) \\ &= |V||I|\cos(\phi)(1 + \cos(2(\omega t + \psi_V))) + |V||I|\sin(\phi)\sin(2(\omega t + \psi_V)) && (2.9) \\ &= P(1 + \cos(2(\omega t + \psi_V))) + Q\sin(2(\omega t + \psi_V)) && (2.10) \end{aligned}$$

here $P = |V||I|\cos(\phi)$ and $Q = |V||I|\sin(\phi)$. We call P *active power* or *real power*, and we call Q *reactive power* or *imaginary power*.

We see that active power and reactive power are dependent on the phase difference ϕ between voltage and current. Furthermore, both terms oscillate with a frequency of $2\omega t$, but the first term $P(1+\cos(2\omega t))$ is unidirectional with an average value of P and the second term $Q(\sin(2\omega t))$ is bidirectional with an average of 0. If $\phi > 0$, then $Q > 0$ and we say that the load consumes reactive power, conversely, if $\phi < 0$, then $Q < 0$ and we say the load supplies reactive power.

We define the *complex power* $S = P + jQ$, and we can calculate this using the voltage and current phasors with the following formula:

$$S = VI^* \tag{2.11}$$

where $\cdot^*$ denotes complex conjugation. The apparent power $|S|$ is the magnitude of the current:

$$|S| = |V||I| = \sqrt{P^2 + Q^2}. \tag{2.12}$$

For complex power and apparent power we use the unit MVA, for active power we use the unit MW and for reactive power we use the unit Mvar.

2.1.3. Impedance and Admittance

In an AC circuit, every component has an impedance, which characterizes the opposition that the component poses to current. The impedance is expressed as a complex number $Z = R + jX$ with unit Ω. Here, the real component R is the resistance and the imaginary component X is the reactance. If a component is inductive with an inductance of L Henry ($L > 0$), we have that $X > 0$ and $jX = j\omega L$. If a component is capacitive with a capacitance of C Farad ($C > 0$), we have $X < 0$ and $jX = 1/j\omega C$. The canonical examples of inductive and capacitive loads are the inductor (or coil) and capacitor.

We call the reciprocal of the impedance the *admittance*, and denote it by Y. The real and imaginary components of the admittance are called the *conductance* and *susceptance*. We write $Y = G + jB$. The conductance and susceptance can be calculated with the following formula:

$$G + jB = (R + jX)^{-1} = \frac{Z^*}{ZZ^*} = \frac{R}{R^2 + X^2} - j\frac{X}{R^2 + X^2}. \tag{2.13}$$

When we view impedance as a complex number and combine this with the phasor notation for voltages and currents, we get a useful extension of Ohm's law for steady state AC-circuits [33, sec. 1.4.1], namely:

$$V = ZI \text{ or } I = YV. \tag{2.14}$$

Here, V is the phasor of the voltage across the component and I the phasor of the current flowing through the component.

Say we have a component with impedance $Z = R + jX$ with $R > 0$, and across this component is a voltage V, which is purely real (i.e., $\psi_V = 0$). Then the current looks as follows:

$$I = YV = V\left(\frac{R}{R^2 + X^2} - j\frac{X}{R^2 + X^2}\right), \tag{2.15}$$

or in polar notation:

$$|I| = \frac{|V|}{R^2 + X^2} \text{ and } \phi = -\operatorname{Arg}(I) = \arctan(X/R) \tag{2.16}$$

Hence, a purely resistive load only influences the amount of current flowing, and the current will be in phase with the voltage (i.e., if $X = 0$ then $\phi = 0$). If, however, the reactance is non-zero, then the current will not be in phase with the voltage. An inductive load, with positive reactance, causes the current to lead the voltage and consumes reactive power (i.e., if $X > 0$, then $\phi > 0$). Conversely, a capacitive load causes the current to lag the voltage and supplies reactive power (i.e., if $X > 0$, then $\phi > 0$). Therefore, an inductive load consumes reactive power and a capacitive load supplies reactive power.

2.1.4. Nominal Voltage and Per Unit Normalization

Each node in an electricity grid is assigned a nominal voltage level, which is the voltage that the node approximately operates at. Typically, the grid consists of a few different nominal voltage levels, and each node is assigned one of the levels. The physical equipment also needs to be rated for this voltage level. The Dutch grid, for example, has four voltage levels in the transmission grid: 110 kV, 150 kV, 220 kV and 380 kV (line-to-line). In practice, voltages will not be exactly equal to the nominal voltages, but will vary slightly around this nominal voltage.

In the study of power systems, it is common to normalize certain numerical values by dividing them by some fixed base quantity [33, sec. 1.8]. The values are then not given in their normal unit, but "per-unit". This normalized quantity is a dimensionless quantity, and we denote it by "pu" (e.g., $|V| = 1.01$ pu). This is often done with voltages, currents, impedances, and powers. Since these quantities are related via (2.11) and (2.14), choosing a base for two of these quantities determines the other two. We choose a base for voltage and power.

For voltages, we select the nominal (line-to-line) voltage as a base, so that in normal operating conditions, we expect the normalized voltage magnitude $|V|$ to be close to 1 pu. This makes it easy to check if the voltages are within normal operating conditions, at a glance, without needing to compare them to the nominal voltage level. For power, we pick the base 1 MVA, as this is a very common unit for power in the transmission grid. This fixes the base for impedances and currents. Suppose the nominal voltage is $\{V^{\text{nom}}\}$ kV, then converting a current from pu to kA can be done by multiplying with $\sqrt{3}\{V^{\text{nom}}\}$, and converting an impedance from pu to Ω can be done by multiplying with $\{V^{\text{nom}}\}^2$. We refer to [33, sec. 1.8] for some examples.

2.1.5. Three Phase Power System

In AC transmission systems, power is usually provided in a three-phase system. However, in a balanced three-phase system, we may instead consider an equivalent single-phase system [33, sec. 1.7]. These networks can be represented in a *one-line diagram*, which only shows a single line per three-phase connection.

Voltages in a three-phase system can be measured either line-to-neutral, or line-to-line. In power systems, the convention is to use line-to-line voltages.

2.2. The Grid as a Graph

We can model the transmission grid as an undirected graph of points in the electrical grid (nodes) that are connected by electrical components (edges). In this section, we describe how the different components in the grid are represented in this graph model.

All points in the grid that satisfy one of the following characteristics, are nodes: all points that directly connect to more than two lines, points that are directly connected to components such as transformers, loads, generators or shunts, or points where specific quantities (e.g., voltage or current flow), are of interest. In the context of transmission grids, nodes are called *buses*, since they usually represent a physical busbar. We will use the terms bus and node interchangeably. The edges of the graph represent connections between the buses, either in the form of high voltage lines or in the form of transformers. Additionally, at each node, we can define a *power injection*. Components such as generators, loads, and shunts are modelled as components that inject power at a specific bus.

We denote the set of all nodes by $\mathcal{N}$, and the set of all edges by $\mathcal{E}$. We denote the graph by $(\mathcal{N}, \mathcal{E})$. Furthermore, we define $N = |\mathcal{N}|$, the total number of nodes in the grid. When we talk about the *topology* of the grid, we mean the graph $(\mathcal{N}, \mathcal{E})$. For example, we might say: "the topology of scenario A and scenario B is the same", which would mean that the graph in the model of scenario A and scenario B is the same[1].

2.2.1. Nodes

In this section, we describe what buses exactly are, the types of buses there are, and the characteristics of these types.

A bus represents a single component in the system. Physically, this point can be a busbar at a substation, but it can also be a (hardwired) three-way junction somewhere in a transmission tower, a place where a DSO or a customer connects to the grid, or a connection to another grid. Furthermore, a physical substation can also include multiple buses, and the number of buses at a substation can depend on the configuration of switches.

Power Injection and Current Injection

Generators and loads that are directly connected to a bus are modelled as *power injections*. The power injection is the complex power that is supplied or consumed by the load or generator and is made up of the active power injection and reactive power injection. The convention is that, when active (or reactive) power is generated, the active (or reactive) power injection is positive, and when it is consumed, the power injection is negative.

If some load or generator injects power at a node, this is of course in the form of a current at a specific voltage. If the injected power is S and the voltage at the node is V (in phasor form), then the *current injection* is the current flowing in or out of the generator or load. It can be calculated via (2.11).

Although shunts can also be considered as a component that injects power, we treat them separately in section 2.2.3.

[1] In this quote, we do not mean that the admittance matrix (see section 3.1) is the same between in scenario A and scenario B.

Voltage and Power Injection

Each bus has four quantities that are of importance. For bus i, these are: total active power injection P_i, total reactive power injection Q_i, voltage magnitude $|V_i|$ and voltage angle δ_i. If multiple loads and generators are attached to the bus, P_i and Q_i are the net generation or consumption of active and reactive power. Since we assume all loads and generators are attached to the bus in parallel, there can only be one bus voltage. The voltage magnitudes are in pu, and the voltage angle is the voltage phase shift (or the angle of the voltage phasor). We can also treat voltage and power injection in their complex form: $V_i = |V_i|e^{j\delta_i}$ and $S_i = P_i + jQ_i$. If these four quantities are known at every bus, the whole state of the system is known.

If the four quantities $P_i, Q_i, |V_i|$ and δ_i are known at each bus i and the admittance of all components is known, then all currents flowing through edges, and currents flowing in and out of buses can be calculated. Therefore, the currents do not need to be known, to specify the full state of the system.

Types of Buses

Depending on the type of bus, some of these quantities can be directly controlled, or are fixed, and are therefore always known. Others are dependent on the interaction with other parts of the grid. For all buses, two of the four quantities are known, and two are unknown.

PQ Bus Loads do not have any control over voltage levels, but their active power and reactive power consumption are known. These types of buses are therefore called *PQ buses*. Depending on how they are connected to the grid, buses that have photovoltaic cells, wind turbines or certain types of generators (so-called *static generators*[2]) attached to them can also behave as PQ buses, so P_i and Q_i are not necessarily negative. A bus with neither generators nor loads attached is modelled as a PQ bus with $P_i = Q_i = 0$.

PV Bus *Synchronous generators*[3] are generators that have control over their output voltage and over their active power output. These buses are therefore called *PV buses*. Any bus that has both loads (i.e. components where the real and reactive power injections are known) and synchronous generators attached to them is also considered as a PV bus.

Slack Bus Due to the conservation of energy, the sum of all power injected plus all power lost (e.g. in transmission lines) should be 0. It is, however, not possible to know the exact amount of losses beforehand. Therefore, we always assign at least one, so-called, *Slack bus* in every grid. This bus is typically a synchronous generator, where we do not fix the power output P_i, but allow it to vary, to make sure this total sum of power is 0. In this way, it "picks up the slack", which explains the name. For a slack bus, the voltage level $|V_i|$ and voltage angle δ_i are fixed. This also defines a reference for voltage angles, based on which all the angles are calculated. In some cases, the slack bus takes the form of a connection to an external grid[4] instead of a synchronous generator.

2.2.2. Edges

Edges are connections between the buses. Physically, these are overhead transmission lines, underground cables, transformers, or (very occasionally) series capacitors or series reactors. All edges have a series impedance, a shunt impedance, and in the case of transformers, also a tap ratio. All edges are modelled via the same model, which is described in appendix A.

Voltage and current over and through a general edge $\{k, l\}$ obey the following equation:

$$\begin{bmatrix} I_{kl}^{\mathrm{f}} \\ -I_{kl}^{\mathrm{t}} \end{bmatrix} = \begin{bmatrix} \frac{1}{|t_{kl}|^2}\left(Y_{kl}^{\mathrm{sr}} + \frac{Y_{kl}^{\mathrm{es}}}{2}\right) & -\frac{Y_{kl}^{\mathrm{sr}}}{t_{kl}{}^{*}t_{lk}} \\ -\frac{Y_{kl}^{\mathrm{sr}}}{t_{kl}t_{lk}{}^{*}} & \frac{1}{|t_{lk}|^2}\left(Y_{kl}^{\mathrm{sr}} + \frac{Y_{kl}^{\mathrm{es}}}{2}\right) \end{bmatrix} \begin{bmatrix} V_k \\ V_l \end{bmatrix}. \tag{2.17}$$

[2]In the rest of this thesis, we use the term *static generator* to refer to any power source that has control over both active and reactive power output.

[3]In the rest of this thesis, we use the term *synchronous generator* to refer to any power source that has control over both active power output and voltage level.

[4]E.g., a connection to the grid of a neighbouring country, or, when only part of a bigger grid is modelled, a connection to the rest of the grid.

Here[5],

$$\begin{aligned} I_{kl}^{\mathrm{f}} &= \text{current flowing out of node } k \\ I_{kl}^{\mathrm{t}} &= \text{current flowing into node } l \\ t_{kl}, t_{lk} &= \text{complex tap transformer ratios} \\ Y_{kl}^{\mathrm{sr}} &= \text{series admittance} \\ Y_{kl}^{\mathrm{es}} &= \text{edge shunt admittance.} \end{aligned}$$

A derivation of this formula can also be found in appendix A. We remark that $I_{kl}^{\mathrm{f}} = -I_{lk}^{\mathrm{t}}$. The complex tap ratio t_{kl} has magnitude τ_{kl}, and angle θ_{kl}. The latter is also referred to as the tap ratio *phase shift*. We have two tap ratios, to model a transformer with a tap on both the high and low voltage side. With t_{kl}, we denote the ratio of the tap on the side of node k, and with t_{lk}, the ratio of the tap on the side of node l. When an edge is a transmission line, $t_{kl} = t_{lk} = 1$. Note that the matrix in (2.17) is only symmetric if t_{kl} and t_{lk} are real.

In a real-life electricity grid, there might be parallel lines or parallel transformers, that connect the same two nodes. However, in our model, we only allow one edge per pair of nodes, as is common for mathematical graphs. When there are parallel lines or transformers, they are combined into a single edge. The matrix in (2.17) for such an edge can then be obtained by summing the admittances of the individual lines together.

Usually, we split I_{kl}^{f} in the following two terms:

$$I_{kl}^{\mathrm{f}} = \left(\frac{V_k}{t_{kl}} - \frac{V_l}{t_{lk}}\right)\frac{Y_{kl}^{\mathrm{sr}}}{t_{kl}^{*}} + V_k \frac{Y_{kl}^{\mathrm{es}}}{2|t_{kl}|^2} \tag{2.18}$$

and indicate the first term by I_{kl}:

$$I_{kl} = \left(\frac{V_k}{t_{kl}} - \frac{V_l}{t_{lk}}\right)\frac{Y_{kl}^{\mathrm{sr}}}{t_{kl}^{*}}. \tag{2.19}$$

2.2.3. Shunts

A shunt is a connection between a bus and ground. Shunts are installed in the power grid by TSO's to either supply or consume reactive power, and can often be connected or disconnected when needed. Just like a generator or load, it injects power at a node. However, unlike a generator or load, the amount of power it injects is always dependent on the voltage at the node. This is why we cannot treat them the same as loads or generators. The shunt typically has a conductance that is (almost) zero, and it is either capacitive or inductive. We denote the admittance of a shunt connected to node i with Y_i^{ns} ("ns" short for "node shunt").

Through a shunt flows the following current, denoted by I_i^{ns}:

$$I_i^{\mathrm{ns}} = V_i Y_i^{\mathrm{ns}}. \tag{2.20}$$

2.2.4. Switches and Breakers

In the physical electricity grid, there are switches and breakers that can connect and disconnect all types of components. In our model, the activation or deactivation of a switch can correspond to the addition or removal of a node or edge to the graph, but it can also have more complex consequences, such as merging nodes or splitting a node into multiple nodes [13, p. 44]. We will not consider switching strategies, but assume that the configuration of all switches and breakers is given.

2.3. Contingencies

During the operation of the transmission grid, sometimes equipment fails. We call this a *contingency*. To assure a stable and uninterrupted operation of the grid, even in the case of contingencies, it is necessary to study how the grid would behave if these contingencies were to occur. Common contingencies that are studied are: the outage of a line or transformer, the outage of a shunt and the outage of a generator.

[5] Here, "f", "t","sr" and "es" are short for "from", "to", "series" and "edge shunt", respectively.

In this report, we assume there is a pre-defined list of contingencies that might occur. Say we have a list of C contingencies, then we index them by the integers $1, \ldots, C$, and we denote the set of all contingencies by $\mathcal{C}$. The base situation, where no contingency has occurred, has index 0. The base situation is also called the *N-0 state*, and when a single contingency has occurred, we are in an *N-1 state*[6]. Here the "N" stands for normal or nominal, and the "-1" means that there is a single contingency.

2.3.1. N-1 Secure

When the grid is operating in its N-0 state, without any unplanned outages, TSO's usually operate their grid in such a way that it is *N-1 secure*. This means that if any single piece of equipment fails, the grid should still be able to operate normally, within all safety margins. For planning and design of the grid, this often means that the grid is designed to be N-2 secure, so that equipment outages can be planned (e.g., for maintenance) all while the grid is still operating in an N-1 secure state.

[6] We can of course extend this to the general *N-n state* for any integer n.

3

Power Flow Modelling

In chapter 2, we introduced equations that describe the behaviour of individual components, and we described how we can model a transmission network as a graph. In this chapter, we first introduce the admittance matrix, then we derive a system of equations that describes the interaction between all components in the grid. In section 3.4, we describe how to solve this system, to obtain an operating state of the grid.

3.1. Admittance Matrix

We now construct the so-called *Admittance Matrix*, Y. This is a complex $N \times N$ matrix, where component Y_{kl} is defined as follows:

$$Y_{kl} = \begin{cases} -\frac{Y^{\mathrm{sr}}_{kl}}{t_{kl}{}^{*} t_{lk}} & \text{if } \{k,l\} \in \mathcal{E} \\ Y^{\mathrm{ns}}_{k} + \sum\limits_{\substack{i \in \mathcal{N} \\ i \neq k}} \frac{1}{|t_{ki}|^2} \left(Y^{\mathrm{sr}}_{ki} + \frac{Y^{\mathrm{es}}_{ki}}{2} \right) & \text{if } k = l \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

If we compare this definition with equation (2.17), we see some similarities. Where the matrix in equation (2.17) describes the voltage and current for a single edge, this matrix combines all the interactions by summing them together[1]. Furthermore, we add the shunt admittance to the diagonal elements.

The admittance matrix is, in general, not symmetric, since Y_{kl} can differ from Y_{lk} if t_{kl} is not real. For large grids, the matrix is usually quite sparse, since the number of elements in row or column i is the number of edges that connect to i. Note that for an edge $\{k,l\}$ in $\mathcal{E}$, equation (2.19) works out to:

$$I_{kl} = Y_{kl}(V_l - V_k). \tag{3.2}$$

This equation actually holds for all distinct pairs of nodes, whether there is an edge connecting them or not because if $\{l,k\} \notin \mathcal{E}$, then $I_{lk} = 0$ and $Y_{kl} = 0$, so it also holds.

3.2. Power Flow Equations

In this section we derive the *power flow equations*, a system of equations relating the four quantities $|V_i|$, δ_i, P_i and Q_i at all nodes $i \in \mathcal{N}$. We mostly follow the reasoning of [13].

[1] To make this precise, we can say that the matrix in (2.17) contains just the four components with indices (k,k), (k,l), (l,k), (l,l), of a larger $N \times N$ matrix, with all other components 0. These larger matrices are then summed together.

3.2.1. Current at a Single Node

We now consider a node k with current injection I_k. By *Kirchhoff's Current Law* (KCL), the injected current is equal to the sum of all current flowing out of the node, either via an edge or via a shunt:

$$I_k = I_k^{\mathrm{ns}} + \sum_{\substack{l \in \mathcal{N} \\ l \neq k}} I_{kl}^{\mathrm{f}}. \tag{3.3}$$

Here I_{kl}^{f} is the current flowing out of node k into edge k, l, if $\{k, l\}$ is an edge (see (2.17)) and $I_{kl}^{\mathrm{f}} = 0$ if $\{k, l\}$ is not an edge. Now we use (2.17) and (2.20):

$$\begin{aligned}
I_k &= I_k^{\mathrm{ns}} + \sum_{\substack{l \in \mathcal{N} \\ l \neq k}} I_{kl}^{\mathrm{f}} && (3.4)\\
&= Y_k^{\mathrm{ns}} V_k + \sum_{\substack{l \in \mathcal{N} \\ l \neq k}} \left[\frac{1}{|t_{kl}|^2} \left(Y_{kl}^{\mathrm{sr}} + \frac{Y_{kl}^{\mathrm{es}}}{2} \right) V_k - \frac{Y_{kl}^{\mathrm{sr}}}{t_{kl}{}^{*} t_{lk}} V_l \right] && (3.5)\\
&= \left(Y_k^{\mathrm{ns}} + \sum_{\substack{i \in \mathcal{N} \\ i \neq k}} \frac{1}{|t_{ki}|^2} \left(Y_{ki}^{\mathrm{sr}} + \frac{Y_{ki}^{\mathrm{es}}}{2} \right) \right) V_k + \sum_{\substack{l \in \mathcal{N} \\ l \neq k}} - \frac{Y_{kl}^{\mathrm{sr}}}{t_{kl}{}^{*} t_{lk}} V_l && (3.6)\\
&= Y_{kk} V_k + \sum_{\substack{l \in \mathcal{N} \\ l \neq k}} Y_{kl} V_l && (3.7)\\
&= \sum_{l \in \mathcal{N}} Y_{kl} V_l && (3.8)
\end{aligned}$$

In the third step we used (3.1). If we define the vectors $\mathbf{I}^{\mathrm{inj}} = (I_1, \ldots, I_N)$ and $\mathbf{V} = (V_1, \ldots, V_N)$, then in matrix notation we can rewrite this to $\mathbf{I}^{\mathrm{inj}} = Y\mathbf{V}$.

3.2.2. Power Injection

Equation (3.8) relates the current injection at each node to the nodal voltages, via the admittance matrix. However, we would like to relate the power injection at each node to the nodal voltages. To do this we take the conjugate of (3.8) and multiply both sides by V_k:

$$\begin{aligned}
S_k &= V_k I_k{}^{*} && (3.9)\\
&= V_k \left(\sum_{l \in \mathcal{N}} Y_{kl} V_l \right)^{*} && (3.10)\\
&= \sum_{l \in \mathcal{N}} V_k V_l{}^{*} Y_{kl}{}^{*} && (3.11)\\
&= \sum_{l \in \mathcal{N}} |V_k||V_l| e^{j(\delta_k - \delta_l)} \left(G_{kl} - jB_{kl} \right) && (3.12)\\
&= \sum_{l \in \mathcal{N}} |V_k||V_l| (\cos(\delta_k - \delta_l) + j \sin(\delta_k - \delta_l)) \left(G_{kl} - jB_{kl} \right). && (3.13)
\end{aligned}$$

Here we split Y_{lk} into its real and imaginary components $Y_{ik} = G_{ik} + jB_{ik}$. Considering the real and complex components separately gives us:

$$\begin{aligned}
P_k &= \sum_{l \in \mathcal{N}} |V_k||V_l| (G_{kl} \cos(\delta_k - \delta_l) + B_{kl} \sin(\delta_k - \delta_l)) && (3.14)\\
Q_k &= \sum_{l \in \mathcal{N}} |V_k||V_l| (G_{kl} \sin(\delta_k - \delta_l) - B_{kl} \cos(\delta_k - \delta_l)). && (3.15)
\end{aligned}$$

3.3. Power Flow Problem Statement

We now state the *Power Flow problem* (sometimes also known as the *Load Flow problem*). For a network with admittance matrix Y, given at every node i two out of the four variables $|V_i|, \delta_i, P_i, Q_i$, find the remaining variables, such that equations (3.14) and (3.15) are satisfied for all k in N. The variables that are known and unknown can be seen in table 3.1.

Bus type	Known variables	Unknown variables
PQ bus	P_i, Q_i	$\lvert V_i\rvert, \delta_i$
PV bus	$\lvert V_i\rvert, P_i$	δ_i, Q_i
Slack bus	$\lvert V_i\rvert, \delta_i$	P_i, Q_i

Table 3.1: Known and unknown variables for node i in the power flow problem, depending on the bus type.

This problem is a nonlinear system of $2N$ equations with $2N$ known, and $2N$ unknown variables. A solution of the Power Flow Problem is sometimes called a *power flow* or a *load flow*.

As we mentioned in section 2.2.1, once the four quantities $|V_i|$, δ_i, P_i and Q_i are known at each bus i, we know the whole state of the system, and the currents through each edge can be calculated from these variables.

In the rest of this thesis, we assume that the stated power flow problems have a unique solution. For more information on this assumption, we refer to [13, ch. 7].

3.3.1. Equivalent Formulations

Note that we have stated the problem here in terms of the voltage magnitude and angle (polar coordinates), and the real and imaginary component of the power injection (Cartesian coordinates). We can, however, also state the problem in terms of complex voltage and power, and complex power injection, or any combination of either Cartesian or polar coordinates of both variables. Furthermore, we could also state the problem in terms of complex voltage and complex current, satisfying (3.8). The most widely used formulation is the formulation as stated here, but for a comparison of the formulations, and their advantages and disadvantages in solving the power flow problem, we refer to [34].

3.4. Power Flow Solvers

We have seen in section 3.2 that finding the state of an electricity grid, equates to solving a nonlinear system of $2N$ equations for $2N$ unknowns. There are multiple numerical algorithms for solving this problem, both for obtaining accurate and approximate solutions. Some of these methods are: the Newton-Raphson method, the Fast Decoupled Load Flow method and DC approximation method [13, ch. 4]. We describe the most common method, via the Newton-Raphson algorithm, in section 3.4.1. We will not treat the DC approximation method here, but there are Optimal Power Flow methods (see chapter 4) that build upon this algorithm, so for a treatment of this method, we refer to [13, sec. 4.3].

3.4.1. Newton Raphson Method

In this section we describe the Netwton Raphson method for solving the power flow problem. We mostly follow the reasoning of [13].

We can solve the system with the Newton-Raphson method. Let δ be a vector containing all unknown voltage angles (i.e., δ_i for all PV- or PQ buses) and let $|\mathbf{V}|$ be a vector containing all unknown voltage magnitudes (i.e., $|V_i|$ for all PQ buses). Let $\mathbf{x}$ be a vector containing both:

$$\mathbf{x} = \begin{bmatrix} \delta \\ |\mathbf{V}| \end{bmatrix}. \tag{3.16}$$

Now we define the active and reactive power-mismatch functions ΔP_i and ΔQ_i as follows:

$$\begin{aligned}
\Delta P_i(\mathbf{x}) &= P_i^{\mathrm{sp}} - P_i && (3.17)\\
&= P_i^{\mathrm{sp}} - \sum_{l\in\mathcal{N}} |V_i||V_l|(G_{il}\cos(\delta_i-\delta_l) + B_{il}\sin(\delta_i-\delta_l)) && (3.18)\\
\Delta Q_i(\mathbf{x}) &= P_i^{\mathrm{sp}} - P_i && (3.19)\\
&= Q_i^{\mathrm{sp}} - \sum_{l\in\mathcal{N}} |V_i||V_l|(G_{il}\sin(\delta_i-\delta_l) - B_{il}\cos(\delta_i-\delta_l)). && (3.20)
\end{aligned}$$

Here, P_i^{sp} and P_i^{sp} are the specified active and reactive power injection, and P_i and Q_i are the computed power injection that follow from (3.14) and (3.15). Note that ΔP_i can only be computed if i is a PV- or PQ bus, and ΔQ_i only if i is a PQ bus, because otherwise, P_i^{sp} or Q_i^{sp} is not given. Now we construct the total power mismatch function as follows:

$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} \Delta\mathbf{P}(\mathbf{x}) \\ \Delta\mathbf{Q}(\mathbf{x}) \end{bmatrix}, \tag{3.21}$$

where $\Delta\mathbf{P}$ and $\Delta\mathbf{Q}$ are the vectors containing all ΔP_i and ΔQ_i that can be computed. We remark that $\mathbf{F}(\mathbf{x}) = 0$, if and only if, $\mathbf{x}$ solves the system for all unknown voltage magnitudes and angles. Furthermore, the remaining unknowns (all of the form P_i or Q_i) can be found by using (3.14) and (3.15). Therefore, we have a solution of the power flow problem, if and only if, we have found $\mathbf{x}$ such that $\mathbf{F}(\mathbf{x}) = 0$. Solving the Power Flow Problem can therefore be done by finding a zero of $\mathbf{F}$.

Finding this zero of $\mathbf{F}$ can be done using the Newton-Raphson algorithm. Each Newton-Raphson iteration then requires solving the following system to obtain the iteration step:

$$-\begin{bmatrix} \frac{\partial\Delta\mathbf{P}}{\partial\delta} & \frac{\partial\Delta\mathbf{P}}{\partial|V|} \\ \frac{\partial\Delta\mathbf{Q}}{\partial\delta} & \frac{\partial\Delta\mathbf{Q}}{\partial|V|} \end{bmatrix} \begin{bmatrix} \Delta\delta \\ \Delta|\mathbf{V}| \end{bmatrix} = \begin{bmatrix} \Delta\mathbf{P} \\ \Delta\mathbf{Q} \end{bmatrix}. \tag{3.22}$$

Here, the left matrix is the Jacobian matrix of $\mathbf{F}$. If we denote the n-th iterate by $\mathbf{x}^n$, then the next iterate is calculated as follows: $\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta\mathbf{x}$, where

$$\Delta\mathbf{x} = \begin{bmatrix} \Delta\delta \\ \Delta|\mathbf{V}| \end{bmatrix}. \tag{3.23}$$

Typically, the iteration is initialized by using a *flat start*. This means that for $\mathbf{x}^0$, all voltages are set to 1 pu and all voltage angles are set to 0. When a solution of an approximate method for solving the Power Flow Problem is known, or a solution of a Power Flow Problem on a very similar network is known, it might also be used as an initial value.

4

Optimal Power Flow Problem

In this chapter, we introduce the *Optimal Power Flow problem* (OPF).

As the name suggests, optimal power flow is about finding some sort of *optimal* way to operate the grid. We have certain assets in the grid that we can control. A specification of how to control a variable of such a component is called a *set-point* (for example, a voltage magnitude set-point for a PV bus). The set-points are our decision variables[1]. Given the set-points, we can perform a power flow to get some state of the system, which needs to satisfy certain constraints. Now we want to find the set-points such that all constraints are satisfied, and do this optimally, which is, of course, characterized by some objective function. We will formulate OPF as a constrained optimization problem.

4.1. Examples of OPF Problems

The OPF problem is not a single specific problem, but a range of problems. Different real life operational scenarios come with different challenges and different available measures to tackle these challenges. When we formulate such a problem as an optimization problem, different challenges are represented by different objective functions, and different available measures are represented by different decision variables or different constraints. To illustrate this, we describe two OPF problems that relate to the two scenarios introduced in section 1.2.

A common problem in OPF literature is the *Economic Dispatch Problem*. This problem is often used as a first example of OPF, see for example [41, Chap. 8]. In this example, the active and reactive power flow demands at all loads are fixed. Each generator has a cost function that is usually a linear, piecewise linear or polynomial function of the active power supplied. The decision variables are the active power, reactive power or voltage set-points for each generator. Typically, there are bounds on the active power output, reactive power output of all generators, the voltage levels at all nodes and the current through each edge. In some versions of the problem, voltage set-points at PV buses are fixed, in others they are also part of the decision variables. In some cases, transformer taps and shunts could also be part of the decision variables, in others they might be fixed or not present. For TenneT, this problem is relevant in a situation where redispatch (see section 1.2.1) is required.

Another OPF problem can be formulated based on the situation described in section 1.2.2. In this formulation, all active power injections are fixed. For all loads, the reactive power injections are fixed and for a few generators, the voltage magnitude (in case of a synchronous generator) or the reactive power output (in case of a static generator) can be controlled. These generators represent the ancillary service providers. TenneT can furthermore control all transformer taps and all controllable shunts. The goal is now to minimize the cost of the ancillary services used, while keeping the voltage magnitudes within safe limits. We call this problem the *Reactive Power Dispatch problem*.

[1]The *decision variables* are the variables that can be directly controlled in the optimization process. In this case, they represent the choices that the grid operator makes.

4.2. Problem Statement

In this section, we state the OPF problem as a general (non-convex) Nonlinear Program (NLP) in the following form:

$$\begin{aligned} \underset{\mathbf{u},\mathbf{x}}{\text{minimize}} \quad & \mathbf{f}(\mathbf{u},\mathbf{x},\mathbf{z}) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{u},\mathbf{x},\mathbf{z}) = 0, \\ & \mathbf{h}(\mathbf{u},\mathbf{x},\mathbf{z}) \leq 0. \end{aligned} \tag{4.1}$$

We mostly follow [26] in this formulation. For now, we assume the decision variables are allowed to vary continuously, but in section 4.4, we will have a closer look at this assumption. We state the problem in the Polar Power-Voltage Formulation, first described in [12] in the 1960s [7, p. 25].

4.2.1. Variables

In (4.1), $\mathbf{u}$ is the vector of decision variables[2]. These can be:

- P_i, when i is a PV bus or a PQ bus
- Q_i, when i is a PQ bus
- $|V_i|$, when i is a PV bus or Slack bus
- τ_{kl} and θ_{kl} when $\{k,l\}$ is a transformer edge with controllable tap ratio
- Y_i^{ns}, when i is a node with a controllable shunt attached

$\mathbf{z}$ is the vector of exogenous variables. These can be:

- P_i and Q_i for each PQ-node i, when it is not a decision variable
- P_i and $|V_i|$ for each PV-node i, when it is not a decision variable
- $|V_i|$ and δ_i, for each slack node i, when it is not a decision variable
- Y_{kl}^{sr} and B_{kl}^{es} for each edge $\{k,l\}$
- Y_i^{ns} for each node i, and τ_{kl} and θ_{kl} for each edge $\{k,l\}$, when it is not a decision variable

$\mathbf{x}$ is the vector of state variables. These can be:

- $P_i, Q_i, |V_i|$ and δ_i for each node i, when it is not a decision variable or exogenous variable

As a general rule, the exogenous variables are the variables that are fixed, the decision variables are the variables that can be directly controlled, and the state variables are variables that cannot be directly controlled, but that can change if the decision variables are changed. Since $\mathbf{z}$ is fixed, we will omit it in the rest of this report. A solution of this problem is some $\mathbf{u}^{\text{opt}}$ that minimizes the objective. Note that for a given decision vector $\mathbf{u}$, we can always obtain the corresponding state $\mathbf{x}$, by performing a Power Flow calculation.

Note that the variables described under decision variables **can** be decision variables, but are not necessarily decision variables. Whether they are decision variables or not depends on the exact scenario that is modelled by the OPF problem (see section 4.1). Remember from section 2.2.1, that a bus without loads or generators attached to it, is modelled as a bus where $P_i = Q_i = 0$.

We remark that for the (ordinary) OPF problem, the distinction between decision variables and state variables is sometimes arbitrary. Take for example those nodes where one of Q_i and $|V_i|$ is a decision variable, and the other a state variable. It does not matter which of the two is the decision variable and which is the state variable. On both state variables and decision variables we can define constraints, and in the end, both are available in the OPF solution. Similarly, we also do not necessarily need a designated slack bus, as long as we have at least one node where the active power output is either a state or a decision variable [7, p. 25]. In Security Constrained OPF, however, this distinction **is** important, as we will see in section 4.3.

[2]We do not specify in what specific order these variables appear as the components of $\mathbf{u}, \mathbf{x}$ and $\mathbf{z}$, as this is not really relevant. We never talk about a component of $\mathbf{u}, \mathbf{x}$ or $\mathbf{z}$ at a specific index. The vectors are more meant as a shorthand for *all variables that are decision/state/exogeneous variables*

Formulation in Terms of Load and Generator Power Injection
In some cases, problem formulations state that voltage set-points and active or reactive power set-points for **generators** or **loads** are controllable, instead of stating that either $|V_i|$, P_i or Q_i is controllable for a **bus**. If all buses have at most one load or generator attached, this makes no difference. If a bus has multiple loads and generators attached, then P_i and Q_i are the net active and reactive power injections. Therefore, we may formulate the problem in terms of generator and load power injections, instead of nodal power injections.

Formulating the problem with load and generator power injections can be beneficial in situations where it is more accurate to define constraints or cost functions for single loads or generators, to accurately model real life situations. These constraints and cost functions could be combined to obtain constraints and cost functions for nodal power injections, but this is not always a trivial task, and can lead to more complex types of constraints or cost functions (e.g., two generators with a linear cost function can have a piecewise linear combined cost function).

We can transform the variables for node i in the following way:

- P_i may be substituted as follows:
$$P_i = \sum_{l \in \mathcal{L}_i} P_l + \sum_{g \in \mathcal{G}_i} P_g \tag{4.2}$$
- Q_i may be substituted as follows:
$$Q_i = \sum_{l \in \mathcal{L}_i} Q_l + \sum_{g \in \mathcal{G}_i} Q_g \tag{4.3}$$
- all synchronous generators that are connected to the same bus, must be operating at the same voltage, so no substitutions for $|V_i|$ or δ_i are made

Here, we denote the set of all generators and loads attached to node i, by $\mathcal{G}_i$ and $\mathcal{L}_i$, respectively, and P_l, P_g, Q_l and Q_g are the active and reactive power injections for a single load or generator. Here again, the power injections are exogenous variables if they are fixed, decision variables if they are controllable, and state variables otherwise.

In the rest of this chapter, we continue to use the formulation in terms of nodal power injections for simplicity of notation. We remark that when the problem is instead formulated in terms of load and generator power injections, the objective and constraints may also be functions of the load and generator power injections.

4.2.2. Equality Constraints
The equality constraints should enforce that all variables together correspond to a valid system, that is, that they should satisfy the non-linear system of equations of the power flow problem. This means that for each node k, equations (3.14) and (3.15) should be satisfied.

4.2.3. Inequality Constraints
The inequality constraints represent physical limits or safety limits on components in the grid. These can be:

- limits on the voltage magnitude at each node
- limits on active or reactive power injections at each node
- limits on current through each edge
- limits on shunt impedance or transformer tap ratios

Often these limits will be simple upper and lower bounds on one of the state or decision variables, but they can also be more complex. For example, the current through each edge is not explicitly included in the decision or state variables, but we can calculate the current flowing through each edge via equation (2.19). An upper bound on the current through an edge $\{k, l\}$ would then look as follows:

$$|Y_{kl}(V_l - V_k)| \leq I_{kl}^{\max} \tag{4.4}$$

for some number $I_{kl}^{\max}$.

4.2.4. Objective Function

The objective function can be a cost function that needs to be minimized, for example, the cost of generators in the economic dispatch problem. In some cases, the objective might also be to minimize losses, carbon emissions or some other measure of performance. We now list some examples.

Minimization of Costs Given a cost per MVA of active or reactive power output for each generator, we can minimize the total cost. Similarly, given an amount of CO_2 emissions per MVA of active or reactive power output for each generator, we can minimize total CO_2 emissions.

Minimization of Power Losses The sum of all active power injections is always non-negative, since the power injections need to compensate for losses in the system. We can minimize the total losses in the system by minimizing $\sum_{i \in \mathcal{N}} P_i$.

Reactive Power Reserve The *reactive power reserve* of a generator is the ability of a generator to supply or consume extra reactive power compared to its operating condition. This reserve generating capacity can be used to maintain the voltage level at a bus, in case extra reactive power supply or consumption is needed due to a contingency. In [28], a *generator reactive margins* term is included in the objective, which looks as follows:

$$\sum_g \left(\frac{Q_g}{Q_g^{\max}} \right)^2 . \tag{4.5}$$

Here the sum is over all generators g, and

$$\begin{aligned} Q_g &= \text{ reactive power output of generator } g \\ Q_g^{\max} &= \text{ maximum reactive power output of generator } g. \end{aligned}$$

By minimizing this objective, the reserve capacity to produce reactive power is maximized. The grid analysis software `PowerFactory` (see section 5.2) includes capabilities to minimize the deviation in reactive power output of generators from either the minimum, the maximum, or some target reactive power output. These are useful (in respective order) in scenarios where critical voltage drops, critical voltage rises, or both might occur [20, sec. 38.2.1.1].

4.3. Security Constrained Optimal Power Flow

In *Security Constrained Optimal Power Flow* (SCOPF), we additionally consider the behaviour of the grid under the influence of contingencies. As we mentioned in section 2.3, it is usually required that the grid is operating in an N-1 secure state, meaning that if any contingency were to occur, all constraints are still satisfied. To ensure this, we need to also look at the state that the system will go to once the contingency has occurred, and see if it still satisfies all limits.

We have noted that we can prepare for certain contingencies (namely those that lead to critical voltage drops or rises) by maximizing reactive power reserve. With this reactive power reserve, we could react to any voltage issues that might occur in case of a contingency. However, such a method does not guarantee that all voltage issues can be resolved, because we never check whether there actually is enough reserve power capacity to resolve all problems. Another issue is that set-points for generators would have to be changed quickly, if a swift restoration of voltage levels is required. This might not always be possible. Furthermore, contingencies could also lead to other problems, such as edge overloading, that cannot be resolved with extra reactive power generation capacity. This means that maximizing reactive power generation is not a complete solution for SCOPF. Instead, we formulate the SCOPF problem as another NLP.

4.3.1. Mathematical Formulation

From now on, we will denote the objective function, equality constraints, and inequality constraints of the (non-security constrained) OPF by $\mathbf{f}_0$, $\mathbf{g}_0$ and $\mathbf{h}_0$, respectively. Given decision variables $\mathbf{u}_0$, we denote the state of the system by $\mathbf{x}_0$. Once a contingency occurs, we essentially obtain a different grid, where the topology of the network might have changed, the admittance matrix might have changed, or power injections might have changed. This new grid also has different constraint functions[3], which we denote by $\mathbf{g}_c$ and $\mathbf{h}_c$ for a contingency c.

We differentiate between two kinds of SCOPF, namely, *preventive* (or *preventative*) and *corrective* SCOPF. We first introduce preventive SCOPF.

4.3.2. Preventive SCOPF

In *preventive SCOPF* we require that for every contingency c, if that contingency occurs, the state of the system still satisfies all constraints $\mathbf{g}_c$ and $\mathbf{h}_c$, with the pre-contingency decision variables $\mathbf{u}_0$. This leads to the following constraints:

$$\mathbf{g}_c(\mathbf{u}_0, \mathbf{x}_c) = 0 \qquad \text{for } c \in \mathcal{C} \tag{4.6}$$

$$\mathbf{h}_c(\mathbf{u}_0, \mathbf{x}_c) \leq 0 \qquad \text{for } c \in \mathcal{C}. \tag{4.7}$$

Note that here $\mathbf{u}_0$ are the same decision variables as the pre-contingency case, but that the state $\mathbf{x}_c$ does change, since a contingency will change the voltages and power injections throughout the grid. In section 4.2.1, we remarked that for ordinary OPF, in some cases, the distinction between decision and state variables is arbitrary, but for SCOPF, this is not the case. A state variable can vary across contingency states, but a decision variable can not.

We now obtain the following optimization problem:

$$\begin{aligned} \underset{\mathbf{u}_0; \mathbf{x}_0, \ldots, \mathbf{x}_C}{\text{minimize}} \quad & \mathbf{f}(\mathbf{u}_0; \mathbf{x}_0, \ldots, \mathbf{x}_C) \\ \text{subject to} \quad & \mathbf{g}_0(\mathbf{u}_0, \mathbf{x}_0) = 0, \\ & \mathbf{h}_0(\mathbf{u}_0, \mathbf{x}_0) \leq 0, \\ & \mathbf{g}_c(\mathbf{u}_0, \mathbf{x}_c) = 0 \quad \text{for } c \in \mathcal{C}, \\ & \mathbf{h}_c(\mathbf{u}_0, \mathbf{x}_c) \leq 0 \quad \text{for } c \in \mathcal{C}. \end{aligned} \tag{4.8}$$

In general, the cost function can be any function of the control variables and all states. An obvious example, is to take the cost function of the N-0 state (i.e., $\mathbf{f}_0(\mathbf{u}_0, \mathbf{x}_0)$), because the grid is most likely to operate in this state. Another example is to take a probability-weighted average of the cost function for all contingency states (i.e., $\sum_{c=0}^{C} p_c \, \mathbf{f}_0(\mathbf{u}_0, \mathbf{x}_c)$, where p_c is the probability that contingency c occurs).

We remark that this formulation leads to a considerable increase in constraints and state variables, compared to a formulation with just the pre-contingency constraints. The number of constraints and state variables are approximately multiplied by $C + 1$. The amount of decision variables does not change. This also means that there might not always be a feasible solution, even if the problem without security constraints has a solution. If, for example, all lines are close to their maximal current rating in the pre-contingency state, and all P_i's are fixed (i.e. we do not allow redispatch), a contingency state where a line fails could be unfeasible.

4.3.3. Corrective SCOPF

In *corrective SCOPF* we allow the decision variables to vary after the occurrence of a contingency, up to a specified amount. This leads to the following optimization problem:

$$\begin{aligned} \underset{\mathbf{u}_0, \ldots \mathbf{u}_C; \mathbf{x}_0, \ldots, \mathbf{x}_C}{\text{minimize}} \quad & \mathbf{f}(\mathbf{u}_0, \ldots, \mathbf{u}_C; \mathbf{x}_0, \ldots, \mathbf{x}_C) \\ \text{subject to} \quad & \mathbf{g}_0(\mathbf{u}_0, \mathbf{x}_0) = 0, \\ & \mathbf{h}_0(\mathbf{u}_0, \mathbf{x}_0) \leq 0, \\ & \mathbf{g}_c(\mathbf{u}_c, \mathbf{x}_c) = 0 \qquad \text{for } c \in \mathcal{C}, \\ & \mathbf{h}_c(\mathbf{u}_c, \mathbf{x}_c) \leq 0 \qquad \text{for } c \in \mathcal{C}, \\ & |\mathbf{u}_0 - \mathbf{u}_c| \leq \Delta\mathbf{u} \qquad \text{for } c \in \mathcal{C}. \end{aligned} \tag{4.9}$$

[3] Many constraints will be the same. Some constraints might no longer be included, (e.g., the contingency is the removal of a line, and therefore the constraint on the loading of that line is no longer relevant). Some constraints might have different bounds (e.g., the contingency is the removal of one of two parallel lines, and therefore the edge can only transport half of the current).

Note that the fact that the decision variables are allowed to change, is reflected in the argument $\mathbf{u}_c$ in the third and fourth constraint. The last constraint restricts by how much the decision variables are allowed to change when a contingency occurs. Some set-points might be allowed to change freely, others might only be allowed to change by a certain amount, or not at all, because of physical limitations on how quickly these set-point could be changed in the case of a contingency.

When the same objective function is used, corrective SCOPF will always find a solution with a better objective value, compared to preventive SCOPF, because the latter is a special case of corrective SCOPF with $\Delta\mathbf{u} = 0$. The number of state variables and constraints in this formulation is comparable to preventive SCOPF, however, the number of decision variables is multiplied by $C + 1$.

4.4. Continuity of Decision Variables

So far, all constraints that we have introduced were continuous. However, in real-world applications, components such as shunts and transformer taps cannot always be controlled continuously, but have a discrete set of possible set-points. This might be binary (i.e., on or off), or a finite amount of possible set-points. In this case, the optimization problem becomes a *Mixed Integer Nonlinear Program* (MINLP). This makes the problem harder to solve. When we have a (SC)OPF problem with discrete variables, its *continuous relaxation* is the problem where all constraints that restrict a single decision variable to a finite or countable set are dropped. In section 5.3.2, we have a look at some ways to deal with discrete variables.

4.5. Other Formulations of the OPF Problem

In this chapter, we have formulated the (SC)OPF problem in terms of active and reactive power and voltage angle and magnitude. This is however not the only way to formulate the problem. It can for example also be formulated with real and imaginary voltages, or with complex currents instead of complex powers [7]. The OPF problem (or slight relaxations of the problem) can also be formulated as a quadratically constrained quadratic program or a semidefinite program [4].

Most scientific literature on SCOPF uses the formulation that we used here [7, sec. 6]. Since set-points and ratings of real-life components are stated in active and reactive power, and voltage magnitude and angle (e.g., specifications for a transformer will provide a range of voltage magnitudes in which it is safe to operate, it will not provide a range for real and imaginary voltage), it is also the formulation that is most easy to apply in a real-world application.

To limit the scope of the research, we stick to the formulation that is used in this chapter.

5

Review of Existing Methods for OPF

In chapter 4, we have formulated the (Security Constrained) Optimal Power Flow problem. The aim of this chapter is to review existing methods for solving the optimal power flow problem, and identify the challenges that come with the problem. We first have a look at two existing software packages for (SC)OPF, and then give an overview of the scientific literature on the problem.

5.1. pandapower

`pandapower` [38] is an open-source software package for the analysis and optimization of power systems. It is entirely written in `Python`, it makes use of `pandas` [37], and it builds upon methods and implementations from `PYPOWER` [27]. `pandapower` includes a way to formulate certain types of OPF problems, and can interface with methods from `PYPOWER` and `PowerModels.jl` [14] for solving these OPF problems. The specifics of the formulation are as follows:

- active and reactive power injections of loads and static generators are controllable
- active power injections of synchronous generators are controllable
- transformers and shunts are not controllable
- cost functions are quadratic polynomials or piecewise linear functions of P_i and Q_i for all controllable components
- upper and lower bounds can be defined on P_i and Q_i for all controllable assets and $|V_i|$ for all buses
- an upper limit can be defined on $|I_{kl}|$ for all edges

We can see that this formulation lacks the ability to have voltage magnitudes as decision variables, however, as we saw in section 4.2.1, there is a workaround for this, by modelling all PV buses as PQ buses. The formulation also lacks control for more advanced components such as shunts or transformers.

When the `PYPOWER` solver is called, it makes use of a primal-dual interior point method, based on the one described in [40]. When the `PowerModels.jl` solver is called, it uses the `Ipopt` [39] solver by default, but it can also use other solvers, such as those of `Gurobi` [21].

5.2. PowerFactory

`PowerFactory` is a commercial software application for modelling and analysis of power systems. It includes an extensive set of network analysis tools, including Power Flow and Optimal Power Flow capabilities. `PowerFactory` is licensed by DIgSILENT GmbH.

The OPF module of `PowerFactory` does not distinguish synchronous generators and static generators, and only allows active and reactive power output as decision variables (see section 4.2.1 on why this can be done). We give an overview of the OPF capabilities of `PowerFactory`:

- active and reactive power output of generators are controllable
- transformer tap positions are controllable (continuous or discrete)
- shunt admittances are controllable (continuous or discrete)
- the following types of objectives can be chosen: minimization of losses, minimization of costs, minimization of load shedding, maximization of reactive power reserve, minimization of control variable deviations
- upper and lower bounds can be set on generator active and reactive power injections and $|V_i|$ for all buses
- an upper limit can be defined on $|I_{kl}|$ for all edges

`PowerFactory` also provides a solver for a simplified version of the model, based on the DC load flow method. In this formulation, a linear approximation is made to the equality constraints. This version is an (Integer) Linear Programming (ILP) formulation and is solved using the simplex method and a branch-and-bound algorithm [20, sec. 25.2.1]. With this simplified model, it is also capable of doing SCOPF.

`PowerFactory` is an application developed for Microsoft Windows, primarily designed to be run on desktop computers. It has an extensive graphical interface, making it suitable for obtaining an overview of large grid models and easy editing. It also has an application programming interface (API) in the programming language `Python`, through which grid data, set-points and results of calculations can be imported and exported.

5.2.1. Downsides of Closed Source Nature of PowerFactory

Since it is not possible to view or modify the source code of `PowerFactory`, it is not as flexible as open-source software. We list some specific issues that TenneT has, that prevent the usage of `PowerFactory` in some parts of its operation.

- Constraints and cost functions for OPF calculations have to be chosen from a specific list and have to be provided in a specific form. This limits the types of OPF calculations that can be performed with the software.
- Performing bulk calculations and working with time series in `PowerFactory` is slow and convoluted. It is not straightforward to do bulk calculations where the grid configuration and power injection scenarios are varied independently.
- There is no access to the details of internal algorithms and the output of intermediate statuses and results in iterative algorithms is limited. This makes it less suited for scientific research because it is hard to make comparisons with other methods.
- When algorithms in `PowerFactory` do not converge, the lack of output and insight into the internal workings of algorithms makes it harder to find the cause of the convergence issues.
- Any additional functionality has to be requested from the manufacturer. There is no guarantee that these features will be implemented, and when they will be available.
- The focus by the manufacturer on desktop use, makes it harder to integrate the software as part of server-based toolchains or automated processes. It does not support the usage on Linux based operating systems, for example.

5.3. Scientific Literature

In this section, we do a literature study on the (SC)OPF problem, focussing on three topics: reducing computational complexity, handling of discrete variables and use of (MI)NLP solvers. We restrict our attention to literature that uses a similar formulation of the (SC)OPF problem as we do.

5.3.1. Reducing Computational Complexity

A lot of literature states that trying to directly solve SCOPF is numerically expensive and that either the size or the number of contingencies has to be drastically limited to obtain a solution in a reasonable time[1] [22, 26, 32]. We quote from [11, sec. 4.1.1]: "The major challenge of the SCOPF is the size of the problem, especially for large systems and [cases] where many contingencies are considered. Trying to solve this problem directly for a large power system, by imposing simultaneously all post-contingency constraints, would lead to prohibitive memory and CPU times requirements." In the literature, a few methods are suggested to reduce the computational complexity. Four main approaches can be observed: contingency filtering, linearization of the problem, problem decomposition, and network compression [11, sec. 4.1].

Contingency Filtering

The idea behind this approach is to only consider *binding constraints*, that is, those constraints that would lead to a better objective value if they were dropped. This is not known beforehand, but contingency filtering techniques exist, that select contingencies that are likely binding, based on some heuristic. The problem is then solved, considering only the selected contingencies. The solution to this problem is then an intermediate solution. Then a power flow is computed with the intermediate solution, for all excluded contingencies, to check if their constraints are met. If this is not the case the process can be repeated with those specific contingencies now also included. A downside to this technique is that the performance can depend on the used heuristic, parameters for this heuristic, and thresholds for in- or excluding contingencies when filtering [11, sec. 4.1.2].

A particular selection procedure, that does not have the drawback of choosing parameters and thresholds, is the so-called *non-dominated contingency* (NDC) procedure. First we define that, given an intermediate solution consisting of decision and state variables $\mathbf{u}$ and $\mathbf{x}$, a contingency a in C is *dominated* by some other contingency b in C, if all constraints that are violated by a, are violated more by b. In the notation of problem (4.8), this is the case if

$$(\mathbf{h}_a(\mathbf{u},\mathbf{x}))_i \leq (\mathbf{h}_b(\mathbf{u},\mathbf{x}))_i \text{ for all } i \text{ where } (\mathbf{h}_a(\mathbf{u},\mathbf{x}))_i > 0. \tag{5.1}$$

Here, $(\mathbf{h}_c(\mathbf{u},\mathbf{x}))_i$ is the i-th component of the vector $\mathbf{h}_c(\mathbf{u},\mathbf{x})$ of some contingency c. How far the constraints are violated is computed by doing a power flow calculation for all contingencies. The idea is that satisfaction of the constraints of contingency b, may also result in the (near) satisfaction of the constraints of a. Therefore, it makes sense to only select contingency b [9]. This selection can be repeated at multiple points in the iteration for different iterates of $\mathbf{u}$ and $\mathbf{x}$.

NDC Filtering is often combined with other methods, for example in [16], where it is combined with another filtering method, and in [22] and [32], where it is combined with Network compression.

Linearization of the Problem

Since linear programs are, in general, much easier to solve than nonlinear programs, some literature suggests transforming the (SC)OPF problem into a linear program. Linearizing the OPF problem involves finding a linear approximation for the cost function $\mathbf{f}$, and the constraint functions $\mathbf{g}$ and $\mathbf{h}$. In [2, sec. 4], two methods for linearizing $\mathbf{g}$ are suggested, one simply linearizes g by use of a first order Taylor approximation, the other method (which is used more often [35, sec. II.C]) does this too, but makes some additional assumptions, leading to a simpler linear system. The second method shares much of the assumptions of the DC power flow method, and is therefore referred to as the DC OPF problem. The linear approach is, however, less suited for situations where accurate modelling of reactive power is important, since the behaviour of reactive power tends to be more nonlinear [35, 8].

Problem Decomposition

In [29], a method for SCOPF with continuous decision variables is introduced, based on the *generalized Benders Decomposition*. It is also used in [26]. In this decomposition, the problem is split into one master problem and a sub-problem for each contingency. In the first iteration, the master problem is just the OPF problem, without any security constraints. First, a solution for this master problem is obtained. Then, for each contingency, a sub-problem is solved. The first step in this sub-problem is checking if the solution of the master problem violates the constraints of that contingency. If that is the case, the second step is to generate a so-called Benders cut. This benders cut is then added as a constraint to the master problem. After all sub-problems have been solved, the cycle repeats itself and the master problem, now

[1] In section 6.2.1, we elaborate on what might be considered as a reasonable time.

with the Benders cuts as added constraints, is solved again. If no more constraints are violated in any of the sub-problems, the process is finished.

This method greatly reduces computation time, compared to obtaining a direct solution of the SCOPF problem. Furthermore, the sub-problems can be solved independent of each other, so an additional speed-up can be obtained by solving them in parallel [26, sec. 4.1.3]. Each sub-problem is roughly as computationally expensive as solving the non-security constrained OPF problem [11, sec. 4.1.3]. However, convergence of the generalized Benders decomposition can only be guaranteed for convex optimization problems, so this method should be used with care [29].

Network Compression
Another method to reduce the computational complexity of the SCOPF problem is the one presented in [22]. It makes use of the fact that the effect of a contingency is usually only felt in a region around the contingency. For each contingency, an active region is identified. This is the region where the contingency has a significant impact on voltages and power flows. They then replace all nodes outside the active region with a so-called *REI-DIMO equivalent network*. This means that the nodes outside the active region are replaced by a reduced number of equivalent nodes, greatly reducing the size of the constraints belonging to that contingency. In numerical experiments, a solution is obtained using this method for a very large model (9241 buses, 12000 contingencies) of the European grid, while having some method to deal with discrete variables, in just over an hour of calculation time [32].

5.3.2. Handling of Discrete Variables

Both in ordinary OPF and SCOPF, discrete variables make the problem significantly harder. For very large SCOPF problems, classical MINLP methods such as branch and bound are incapable of obtaining solutions in reasonable time frames [11, sec. 4.3]. Another complication is that some state-of-the-art MINLP solvers do not support trigonometric functions [1]. In [17, sec. IV], it is mentioned that solving the OPF and SCOPF problem directly with MINLP solver `Bonmin` was attempted for a grid with 60 nodes and 33 contingencies. The solver found a solution for the OPF problem, but did not provide a feasible solution for the SCOPF problem after several hours of running. Instead of MINLP methods, some algorithms use continuous relaxations of the problem, and then use rounding techniques to obtain solutions that are feasible for the discrete problem. These techniques cannot guarantee optimality of the solution, however.

A simple rounding method for mixed integer OPF problems is described in [25, sec. 2.1]. First, a solution of the continuous relaxation of the OPF is obtained. Then all variables that are required to be discrete, are rounded to their nearest discrete value. The OPF problem is then solved again with the discrete variables fixed to their rounded of values, the continuous decision variables are now the only decision variables.

In [23, sec. II.D] a progressive round off method is suggested for SCOPF. This method first rounds off those variables that are nearest to a discrete value. Then another round of optimization is done with the rounded off variables fixed, and all other variables still considered as continuous. Then, again, the variables that are closest to a discrete value are rounded off. This is iterated until all values are rounded off. This method is also used in [32].

Another method is introduced in [25, sec. 3.3], the so-called *objective feasibility pump*. Once again, first a solution is obtained of the continuous relaxation. Then, for each variable that is required to be discrete, the distance of that variable to its closest discrete value is added to the objective as a penalty term. These terms are called the *feasibility pump*. Now an iterative process starts, where in each iteration the OPF with feasibility pump term added is solved, and after each iteration the weight of the feasibility term is increased. The process halts when a solution is obtained that satisfies all discrete constraints (up to a certain tolerance).

The method proposed in [17] uses a similar approach as the objective feasibility pump. First, it converts every discrete variable into a set of binary variables. Then it relaxes the problem to a continuous problem and adds a penalty term to the objective for every binary variable. The penalty term is based on the *Fischer Burmeister function* and it penalizes the binary values for being away from zero or one. In numerical experiments [17, sec. IV], the proposed method outperforms the simple rounding method, both in finding feasible solutions and in finding more optimal solutions.

A downside to the latter three methods is that they all have one or multiple parameters that require tuning. For the progressive round off method this is the bin size of which variables are considered close enough to be rounded off each iteration. In the object feasibility pump method it is the initial weight of the penalty terms. In the last method, it is a parameter for the Fischer Burmeister function and a weight for the penalty terms.

5.3.3. (MI)NLP Solvers and Algorithms

The challenge of finding solutions of (MI)NLP problems is not exclusive to optimal power flow. Therefore, a lot of literature makes use of existing solvers, or algorithms proposed in other literature, to obtain solutions. These are some of the solvers that can be found in the literature:

- [16, 25, 17] use the `Ipopt` solver [39], an open-source software package for large-scale nonlinear optimization. It implements an interior point line search filter method. `PowerFactory` and `pandapower` also have the option to use `Ipopt`.
- The `PYPOWER` OPF solver is a primal-dual interior point method based on [40].
- [32] makes use of a solver based on [5], a primal-dual interior point method using projected conjugate gradient iteration.
- [23] uses the commercial solver `Knitro` [6], which implements multiple algorithms, the two relevant ones are both primal-dual interior point methods, one of them is based on [5].
- [26] uses a `MATLAB` solver.
- [9] refers to [10], which compares three interior point methods for OPF problems. It recommends a *predictor-corrector* and a *multiple centrality corrections* based interior-point method.
- [17] mentions that the `Bonmin` [15] solver is able to solve non-security constrained OPF problems, but is not able to solve SCOPF problems. `Bonmin` is an experimental open-source MINLP solver for general MINLP problems.

5.3.4. Optimality of Solutions

Not many claims are made about the optimality of the solutions that different methods obtain. Some claims are made about local optimality for continuous relaxations of the SCOPF problem [32], but to the author's knowledge, no methods claim to be able to find globally optimal solutions, for all but very small-scale problems. We quote from [36, sec. 2.3]: "It is unlikely that useful theories of convergence or global optimality can be developed for non-trivial real-life OPF problem formulations." Instead, the focus lies on reliably obtaining feasible solutions, with acceptable objective values.

5.4. Conclusion

Both in literature and in commercial software, methods have been suggested and implemented to perform (SC)OPF. It is, however, still an active area of research and no standard method exists that is both fast and accurate and always finds globally optimal solutions for security constrained OPF with discrete decision variables.

There are existing open-source methods, such as `pandapower`, but the capabilities and configurability of these methods is quite limited. Commercial solutions such as `PowerFactory` have more capable (SC)OPF methods, but their closed-source nature restricts their applicability to a specific set of OPF problem types because cost functions and constraints can only be configured in a particular form. Furthermore, implementing (SC)OPF calculations as part of an automated process, or as part of bulk calculations, where (SC)OPF calculations are performed on large amounts of different scenarios and grid variations, can be difficult with commercial solutions, since the configuration options of the software can be limited.

From the scientific literature, it is clear that solving large scale (SC)OPF problems has two major challenges: high computational costs, and treatment of discrete variables. On both of these challenges, a lot of research has been done, and some methods for dealing with these challenges have been suggested. Although some comparisons have been made between different methods, there is no wide-spread consensus in the literature about which of these methods perform the best. In most cases, the methods involve an iteration process for checking contingency states, dealing with violated constraints in contingency states, possibly dealing with discrete variables and obtaining a new solution. Part of this iteration process is also solving a (non-)linear program, which is often done using interior point methods, either via existing solvers or by implementing algorithms found in the literature. Some methods in the literature are capable of doing large scale SCOPF in reasonable timeframes, with some way of dealing with discrete variables. In general, no claims are made about the global optimality of solutions obtained by any of the methods.

6 Research Goals

In this chapter, we start by sketching the context in which the research is taking place. Then we formulate a research question, and elaborate on the requirements.

6.1. Research Context

At TenneT, there is a need for tools that can do fast, robust, and accurate network analysis calculations, such as power flow calculations, analysis of the grid in contingency situations, and optimal power flow calculations. These calculations sometimes need to be done as individual calculations, and sometimes in bulk, calculating for hundreds of timestamps or different scenarios, or as part of automated processes. For some types of calculations, commercial software, such as `PowerFactory` satisfies all needs. However, `PowerFactory` is not suited for all uses cases, as is illustrated by the examples in section 5.2.1.

In this context, the `ODINA` toolbox is being developed at TenneT. This is a toolbox for transmission network analysis. The aim of the toolbox is to implement fast and robust network calculations, that can be used in automated processes. The toolbox can interface with `PowerFactory`, so that existing network models that are made in `PowerFactory` can be imported into the toolbox for calculations. So far, the toolbox is capable of doing fast power flow calculations, with a few different implemented algorithms, in bulk. The `ODINA` toolbox provides an environment that is well suited for the implementation of a (SC)OPF algorithm, since it provides methods to import grid models, calculate admittance matrices, and perform power flow calculations.

6.2. Research Question

Our research question is:

> Can we implement a (SC)OPF algorithm, with a focus on speed, robustness, and configurability, that works for the Dutch transmission grid?

We elaborate on the requirements on speed, robustness, and configurability.

6.2.1. Requirements on Speed

In operation of the transmission grid, (Security Constrained) Optimal Power Flow could be used for calculating optimal set-points for controllable components. In order for the result of (SC)OPF calculations to be used in real-time operation of the grid, calculations have to be done in a limited timespan. Predictions of power generation and consumption are not always known very far in advance, and reconfiguration of the grid is done on an hourly basis. Therefore, calculations on the full grid, for 24 hourly timestamps, should ideally not take longer than an hour. If this is not feasible, bottlenecks should be identified and suggestions should be made to further improve speed.

In planning of the grid, (SC)OPF could be used by analysing possible configurations of the grid, and calculating associated operating costs for many operation scenarios. To be able to effectively do these kinds of analyses, (SC)OPF calculations should be able to be performed in bulk, making use of techniques such as parallelization and warm starting, whenever possible.

6.2.2. Requirements on Robustness

An important condition for the implementation of (SC)OPF calculations in business processes of TenneT, is that the algorithms should be robust. The methods should work on a broad range of grid variations and operating scenarios. Most importantly, it should always find feasible solutions, and although (proven) strict optimality might not be realistic, results should always be near optimal. In the case that it cannot be prevented that the method fails in some cases, it should clearly indicate to the user what went wrong and suggest solutions. These requirements on robustness are focussed on models of the Dutch grid. Robust convergence on grid models that are not similar to the Dutch grid would be an added benefit, but this is of secondary importance.

6.2.3. Requirements on Configurability

As described in chapter 4, (SC)OPF is a broad category of problems, and it can vary between applications what the decision variables and constraints are and what the objective is. The aim is to make an (SC)OPF module that is easy to configure for a broad range of types of (SC)OPF problems, so that it is widely applicable. Furthermore, it should support bulk calculations, where calculations are done with different power injection scenarios, grid topology scenarios, and contingency scenarios. The support of configuration options may be restricted to use cases that are specific to the Dutch grid.

6.3. Size of Used Models

TenneT has a full model of the Dutch high voltage grid that is used in operation and planning of the grid. Some details of the model can be found in table 6.1. For the contingencies, we will consider the outage of every single transformer and every single line.

Component	Number	Number controllable
Nodes	1636	-
Edges	1586	-
Lines	994	-
Transformers	592	280
Shunts	132	132

Table 6.1: Overview of the size of the model of the Dutch transmission grid.

6.3.1. Size of Test Models

During development and testing, we use some smaller scale models. Here is an overview of the used models:

- the 14 bus example [18] included in `PowerFactory`
- the 39 bus example [19] included in `PowerFactory`
- the Nordic test system [31]

We may modify the grids, by adding tap controls to transformers, or controllable shunts to buses, to be able to test problems with tap controls and shunt controls as decision variables. An overview of the size of the models can be found in table 6.2.

Model	Number of nodes	Number of edges
14 Bus model	14	21
39 Bus model	39	46
Nordic test system	74	102

Table 6.2: Overview of the size of the test models.

References

[1] Kevin-Martin Aigner et al. "Solving AC Optimal Power Flow with Discrete Decisions to Global Optimality". In: *INFORMS Journal on Computing* 35.2 (2023). Cited by: 4, pp. 458–474. DOI: 10.1287/ijoc.2023.1270.

[2] O. Alsac et al. "Further Developments in LP-Based Optimal Power Flow". In: *IEEE Transactions on Power Systems* 5.3 (1990), pp. 697–711. DOI: 10.1109/59.65896.

[3] TenneT TSO B.V. *Grid Map Onshore Netherlands EN*. 2024. URL: https://www.tennet.eu/grid/grid-maps (visited on 12/12/2024).

[4] Daniel Bienstock et al. "Mathematical Programming Formulations for the Alternating Current Optimal Power Flow Problem". In: *4OR* 18 (Sept. 2020), pp. 249–292. DOI: 10.1007/s10288-020-00455-w.

[5] Richard H Byrd, Mary E Hribar, and Jorge Nocedal. "An interior point algorithm for large-scale nonlinear programming". In: *SIAM Journal on Optimization* 9.4 (1999), pp. 877–900.

[6] Richard H Byrd, Jorge Nocedal, and Richard A Waltz. *Knitro: An Integrated Package for Nonlinear Optimization*. 2005.

[7] Mary B Cain, Richard P O'neill, Anya Castillo, et al. "History of Optimal Power Flow and Formulations". In: *Federal Energy Regulatory Commission* 1 (2012), pp. 1–36.

[8] Florin Capitanescu. "Critical Review of Recent Advances and Further Developments Needed in AC Optimal Power Flow". In: *Electric Power Systems Research* 136 (July 2016), pp. 57–68. ISSN: 03787796. DOI: 10.1016/j.epsr.2016.02.008.

[9] Florin Capitanescu et al. "Contingency Filtering Techniques for Preventive Security-Constrained Optimal Power Flow". In: *IEEE Transactions on Power Systems* 22 (4 Nov. 2007), pp. 1690–1697. ISSN: 08858950. DOI: 10.1109/TPWRS.2007.907528.

[10] Florin Capitanescu et al. "Interior-point based algorithms for the solution of optimal power flow problems". In: *Electric Power Systems Research* 77 (5-6 Apr. 2007), pp. 508–517. ISSN: 03787796. DOI: 10.1016/j.epsr.2006.05.003.

[11] Florin Capitanescu et al. "State-of-the-Art, Challenges, and Future Trends in Security Constrained Optimal Power Flow". In: *Electric Power Systems Research* 81 (8 2011), pp. 1731–1741. ISSN: 03787796. DOI: 10.1016/j.epsr.2011.04.003.

[12] Jacques Carpentier. "Contribution a l'Etude du Dispatching Economique". In: *Bull. Soc. Fr. Elec. Ser.* 3 (1962), p. 431.

[13] Shravan Chipli. "Automating AC Power Flow Simulations". MA thesis. TU Delft, Aug. 2021.

[14] Carleton Coffrin et al. "PowerModels.jl: An Open-Source Framework for Exploring Power Flow Formulations". In: *2018 Power Systems Computation Conference (PSCC)*. June 2018, pp. 1–8. DOI: 10.23919/PSCC.2018.8442948.

[15] COIN-OR. *Basic Open-source Nonlinear Mixed INteger programming*. URL: https://www.coin-or.org/Bonmin/ (visited on 12/18/2024).

[16] Frank E. Curtis et al. *A Decomposition Algorithm for Large-Scale Security-Constrained AC Optimal Power Flow*. 2021. URL: https://arxiv.org/abs/2110.01737.

[17] Elnaz Davoodi and Florin Capitanescu. "A Robust Penalty-Based Approach to Optimal Reactive Power Dispatch With Discrete Controls". In: *2023 IEEE Belgrade PowerTech*. 2023, pp. 1–7. DOI: 10.1109/PowerTech55446.2023.10202758.

[18] DIgSILENT GmbH. *14 Bus System. DIgSILENT PowerFactory*.

[19] DIgSILENT GmbH. *39 Bus New England System. DIgSILENT PowerFactory*.

[20] DigSILENT GmbH. *DIgSILENT PowerFactory 2022, User Manual*. Aug. 2022.

[21] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. 2024.

[22] Karim Karoui et al. "Large Scale Security Constrained Optimal Power Flow". In: *2008 Power Systems Computation Conference (PSCC)*. July 2008.

[23] Karim Karoui et al. "New Large-Scale Security Constrained Optimal Power Flow Program Using a New Interior Point Algorithm". In: *2008 5th International Conference on the European Electricity Market*. 2008, pp. 1–6. doi: 10.1109/EEM.2008.4579069.

[24] James L Kirtley. *Electric Power Principles: Sources, Conversion, Distribution, and Use*. Wiley, 2010. isbn: 9780470667170.

[25] Peijie Li, Jianming Su, and Xiaoqing Bai. "An Objective Feasibility Pump Method for Optimal Power Flow with Unit Commitment Variables". In: *Electric Power Systems Research* 236 (Nov. 2024). issn: 03787796. doi: 10.1016/j.epsr.2024.110928.

[26] Yuan Li et al. "Decision Making Under Uncertainty in Power System Using Benders Decomposition". PhD thesis. Iowa State University, 2008.

[27] Richard Lincoln. *PYPOWER*. 2023.

[28] Enrique Lobato Miguélez, Francisco M. Echavarren Cerezo, and Luis Rouco Rodrguez. "On the Assignment of Voltage Control Ancillary Service of Generators in Spain". In: *IEEE Transactions on Power Systems* 22 (1 Feb. 2007), pp. 367–375. issn: 08858950. doi: 10.1109/TPWRS.2006.888984.

[29] A Monticelli and M V F Pereira. "Security-Constrained Optimal Power Flow with Post-Contingency Corrective Rescheduling". In: *IEEE Transactions on Power Systems* (1 1987).

[30] Peter van Oirsouw and JFG Cobben. *Netten voor distributie van elektriciteit*. Phase to Phase, 2011.

[31] Luis David Pabón Ospina, Andrés Felipe Correa, and Gustav Lammert. "Implementation and validation of the Nordic test system in DIgSILENT PowerFactory". In: *2017 IEEE Manchester PowerTech*. 2017, pp. 1–6. doi: 10.1109/PTC.2017.7980933.

[32] Ludovic Platbrood et al. "A Generic Approach for Solving Nonlinear-Discrete Security-Constrained Optimal Power Flow Problems in Large-Scale Systems". In: *IEEE Transactions on Power Systems* 29 (3 2013), pp. 1194–1203.

[33] Pieter Schavemaker and Lou van der Sluis. *Electrical Power System Essentials*. John Wiley and Sons, Incorporated, 2017. isbn: 9781118803455.

[34] Baljinnyam Sereeter, Cornelis Vuik, and Cees Witteveen. "On a Comparison of Newton–Raphson Solvers for Power Flow Problems". In: *Journal of Computational and Applied Mathematics* 360 (Nov. 2019), pp. 157–169. issn: 0377-0427. doi: 10.1016/J.CAM.2019.04.007.

[35] Brian Stott, Ongun Alsac, and Alcir J Monticelli. *Security Analysis and Optimization*. Dec. 1987.

[36] Brian Stott and Ongun Alsaç. "Optimal Power Flow - Basic Requirements for Real-Life Problems and their Solutions". In: 2012.

[37] The pandas development team. *pandas-dev/pandas: Pandas*. Version 2.14.10. Feb. 2020. doi: 10.5281/zenodo.3509134.

[38] L. Thurner et al. "pandapower — An Open-Source Python Tool for Convenient Modeling, Analysis, and Optimization of Electric Power Systems". In: *IEEE Transactions on Power Systems* 33.6 (Nov. 2018), pp. 6510–6521. issn: 0885-8950. doi: 10.1109/TPWRS.2018.2829021.

[39] Andreas Wächter and Lorenz T. Biegler. "On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming". In: *Mathematical Programming* 106 (1 May 2006), pp. 25–57. issn: 00255610. doi: 10.1007/s10107-004-0559-y.

[40] Hongye Wang et al. "On Computational Issues of Market-Based Optimal Power Flow". In: *IEEE Transactions on Power Systems* 22 (3 Aug. 2007), pp. 1185–1193. issn: 08858950. doi: 10.1109/TPWRS.2007.901301.

[41] Allen J Wood, Bruce F Wollenberg, and Gerald B Sheblé. *Power Generation, Operation, and Control*. Third edition. Wiley-IEEE, 2013. isbn: 9781118733912.

A

Admittance Matrix for General Edge

In this chapter, we derive the admittance matrix for a general edge $\{k, l\} \in \mathcal{E}$ that is given in (2.17).

All transmission lines in the grid, whether overhead lines or underground cables, are modelled via the *standard pi-model* for a transmission line. For the details of this model, we refer to [33, app. E]. We model transformers as a pi-model line with an ideal transformer in series at one or both ends, as is shown in Figure A.1. If we set the transformer taps $t_{kl} = t_{lk} = 1$, we obtain back the standard-pi model. Therefore, the model in Figure A.1 can model both lines and transformers and is thus our model for a general edge. Although we do not use the exact same, model information on modelling transformers can be found in [33, app. B]. The transformers could also be modelled as a single ideal transformer in series with a pi-model line, however, since a lot of physical tap transformers have a tap on both the low voltage and high voltage side, it is easier to model them with two ideal transformers. In this way, we can control two tap ratios, just like for a physical transformer.

Note that when we are working in a per unit normalized system, the ratio is also normalized. If we have, for example, a transformer between the buses k and l that are at their nominal voltages 110 kV and 220 kV, respectively, then the non-normalized tap ratio is $t_{kl} = 1/2$, but the normalized tap ratio is $t_{kl} = 1$.

Figure A.1 contains all variables in (2.17). Additionally, currents $I_1, \dots, I_5$ are introduced for use in the derivation in this chapter (even though they share the notation, they are not related to current injections at nodes). Furthermore, we have also introduced two temporary nodes, k' and l', for the derivation. The voltages across these nodes are denoted by voltage phasors $V_{k'}$ and $V_{l'}$, respectively.

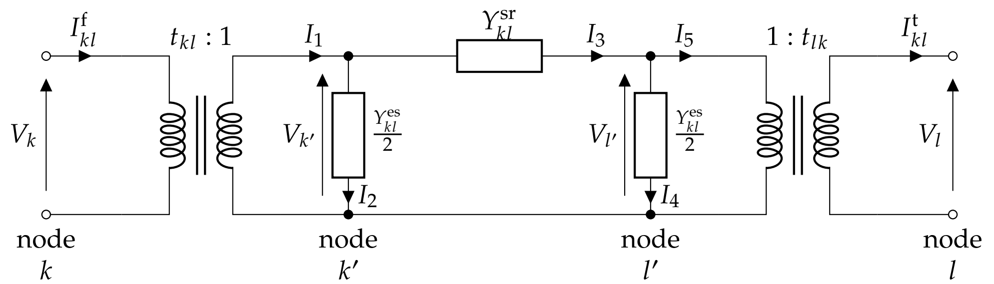

Figure A.1: Model for general edge, with two ideal transformers in series with a standard pi-model line. Admittances are shown for each load.

We note that voltages and currents on both sides of the transformers are related via the transformer tap ratios t_{kl} and t_{lk}:

$$V_k = t_{kl} V_{k'} \tag{A.1}$$
$$V_l = t_{lk} V_{l'} \tag{A.2}$$
$$I^{\mathrm{f}}_{kl} = \frac{I_1}{{t_{kl}}^*} \tag{A.3}$$
$$I^{\mathrm{t}}_{kl} = \frac{I_5}{{t_{lk}}^*}. \tag{A.4}$$

Now, KCL gives us the following:

$$I_1 = I_2 + I_3 \tag{A.5}$$
$$I_5 = I_3 - I_4. \tag{A.6}$$

For the shunt capacitance of the line, we use Ohm's law:

$$I_2 = V_{k'} \frac{Y^{\mathrm{es}}_{kl}}{2} = \frac{V_k}{t_{kl}} \frac{Y^{\mathrm{es}}_{kl}}{2} \tag{A.7}$$
$$I_4 = V_{l'} \frac{Y^{\mathrm{es}}_{kl}}{2} = \frac{V_l}{t_{lk}} \frac{Y^{\mathrm{es}}_{kl}}{2}. \tag{A.8}$$

And similarly for the current I_3:

$$I_3 = Y^{\mathrm{sr}}_{kl}(V_{k'} - V_{l'}) = Y^{\mathrm{sr}}_{kl} \left(\frac{V_k}{t_{kl}} - \frac{V_l}{t_{lk}} \right). \tag{A.9}$$

Now if we combine (A.3), (A.5), (A.7) and (A.9), we get:

$$I^{\mathrm{f}}_{kl} = \frac{1}{{t_{kl}}^*} \left(\frac{V_k}{t_{kl}} \frac{Y^{\mathrm{es}}_{kl}}{2} + Y^{\mathrm{sr}}_{kl} \left(\frac{V_k}{t_{kl}} - \frac{V_l}{t_{lk}} \right) \right) \tag{A.10}$$
$$= \frac{1}{|t_{kl}|^2} \left(Y^{\mathrm{sr}}_{kl} + \frac{Y^{\mathrm{es}}_{kl}}{2} \right) V_k - \frac{Y^{\mathrm{sr}}_{kl}}{{t_{kl}}^* t_{lk}} V_l. \tag{A.11}$$

Similarly, we combine (A.4), (A.6), (A.8) and (A.9), to obtain:

$$I^{\mathrm{t}}_{kl} = \frac{1}{{t_{lk}}^*} \left(Y^{\mathrm{sr}}_{kl} \left(\frac{V_k}{t_{kl}} - \frac{V_l}{t_{lk}} \right) - \frac{V_l}{t_{lk}} \frac{Y^{\mathrm{es}}_{kl}}{2} \right) \tag{A.12}$$
$$= -\frac{1}{|t_{lk}|^2} \left(Y^{\mathrm{sr}}_{kl} + \frac{Y^{\mathrm{es}}_{kl}}{2} \right) V_l + \frac{Y^{\mathrm{sr}}_{kl}}{t_{kl} {t_{lk}}^*} V_k. \tag{A.13}$$

Writing this as a matrix vector product gives us equation (2.17).