CHAIR FOR ELECTRICAL ENEGY SYSTEMS
Chairman: Univ.-Prof. Dr.-Ing. Matthias Luther
Univ.-Prof. Dr.-Ing. Johann Jäger

Master Thesis M347

Modeling of Fast-Switching Transformers for Voltage Stability Studies in Python

Author: Maximilian Markus Veit Köhler, B.Eng.
23176975

Supervisor: Ilya Burlakin, M.Sc.
Georg Kordowich, M.Sc.

Date of submission: 02nd May 2025

[March 10, 2025 at 17:02 – preliminary version 0.1]

Contents

1 Introduction

Some blibla as introduction. [**machowskiPowerSystemDynamics2020**]

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris. Some more Text.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa. [MK]Write a nice introduction.

Research Interests

Here are gaps and possible extension of knowledge.

Here are the research objectives and questions.

- Influence of OLTC control on possible operational uses: Short-term voltage stability, long-term voltage stability;
- Can a increased dynamic regulation help machine recovery?
- Does the increased tap ratio gradient harm transient stability of machines? Does it help or harm CCT of machines or machine groups?
- Transformers act as big low-pass filters: Can this behavior be beneficial as well for the interactions of inverters in the grid on AC side (in the sense of Harmonic Stability)? **[Quelle]**

Research Question of this Thesis

How do different control types and characteristics of Tap Changing transformers influence the voltage stability of the given system?

Therefore following questions/steps can be imagined as supportive:

1. How can Voltage stability of a system be classified and be looked at? Which indices, measurements, etc.
2. Which transformer model has to be considered to show influences?
3. Which additional load models, source models, transmission model have to be modeled for an adequate assessment?
4. Which systems are useful to consider in showing effects? Which circumstances lead to a stability support, which to a decrease? Where can limits be drawn?

Additionally during the process of the thesis, the following question came up as an extension. Is is the second interest of this thesis, and shall be more focused in the later part. Therefore some assessments in the **??** are conducted.

Additional Question of this Thesis

Can the already existing Tap Changer Control of the **FSM!** (**FSM!**) be improved towards a more operation oriented control?

Construction of the Thesis

2 Fundamentals

Following chapter shall introduce the basics for implementing an **OLTC!** equipped transformer into a existing **PSS!** framework. This is considering the already existing surrounding, more detailed the electric behavior of the transformer itself and some control engineering theory for the corrosponding **OLTC!**. Thus its main goal is increasing voltage stability [**machowskiPowerSystemDynamics2020**], main indices and assessment methods are considered as well.

2.1 Power System Modeling

Transformer Π-Model:

- Deviation from 'normal' circuit
- Placing the impedance elements on the LV side vs. HV side
- Getting into a single line Π-model
- Introducing the tapping factor ϑ; if phase shifting it is getting complex -> how to consider this
- Different Y-Bus matrices if LV or HV side is considered
- Per-unit problems with the transformer characteristics
- Short overview of Open-Source power system simulation packages

Voltage stability:

- Definitions, classifications, conditions
- PV-curves / QV-curves
- Derivation of Indices: Practical application

Control engineering:

- State-of-the-art control mechanisms for OLTCs
- Chaos theory controllers

2.1.1 General and Existing Model

2.1.2 Transformer Electric Model and Behavior

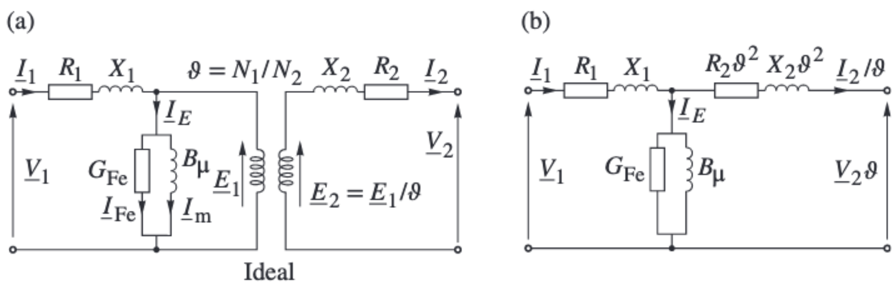

Figure 2.1: Two-Winding Transformer Circuit in the Positive Sequence; a) ideal representation with impedances on each **HV!** and **LV!** side and b) related impedances on the XX side; own figure after [**machowskiPowerSystemDynamics2020**]

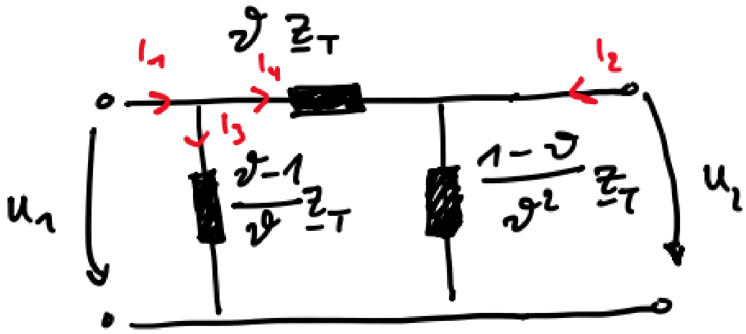

Figure 2.2: Π-representative circuit of a transformer with a longitudinal tap changer; own figure after [**machowskiPowerSystemDynamics2020**, **burlakinEnhancedVoltageControl2024**]

$$\underline{I} = \mathbf{Y} \cdot \underline{U}$$

$$\begin{bmatrix} \underline{I}_1 \\ \underline{I}_2 \end{bmatrix} = \begin{bmatrix} \underline{Y}_{11} & \underline{Y}_{12} \\ \underline{Y}_{21} & \underline{Y}_{22} \end{bmatrix} \cdot \begin{bmatrix} \underline{U}_1 \\ \underline{U}_2 \end{bmatrix} \tag{2.1}$$

The admittance matrix of a two port network can be expressed after **machowskiPowerSystemDynamics** as **??**. For the Π-model of an **OLTC!** transformer it is leading to **??**.

$$\underline{\mathbf{Y}}_{\Pi,\mathrm{T}} = \begin{bmatrix} \underline{Y}_{\mathrm{T}} & -\underline{\vartheta}\underline{Y}_{\mathrm{T}} \\ \underline{\vartheta}^{*}\underline{Y}_{\mathrm{T}} & -\underline{\vartheta}^{*}\underline{\vartheta}\underline{Y}_{\mathrm{T}} \end{bmatrix} \tag{2.2}$$

Another way of writing down the admittance matrix is shown in **??**. It is considering, that the matrix can be split up in a symmetric, constant part, and a variable current injection part. The latter is not symmetrical and depends on the tap position of the transformer. Therefore in some simulation algorithms the static part is used in the admittance matrix, and the variable part is considered in the current injection vector. [**machowskiPowerSystemDynamics2020**]

$$\begin{bmatrix} \underline{I}_1 \\ -\underline{I}_2 \end{bmatrix} = \begin{bmatrix} \underline{Y}_{\mathrm{T}} & -\underline{Y}_{\mathrm{T}} \\ -\underline{Y}_{\mathrm{T}} & \underline{Y}_{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \underline{U}_1 \\ \underline{U}_2 \end{bmatrix} - \begin{bmatrix} \Delta\underline{I}_1 \\ \Delta\underline{I}_2 \end{bmatrix} \text{, where}$$

$$\begin{bmatrix} \Delta\underline{I}_1 \\ \Delta\underline{I}_2 \end{bmatrix} = \begin{bmatrix} \underline{0} & (\underline{\vartheta} - 1)\underline{Y}_{\mathrm{T}} \\ -(\underline{\vartheta}^{*} + 1)\underline{Y}_{\mathrm{T}} & (\underline{\vartheta}^{*}\underline{\vartheta} + 1)\underline{Y}_{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \underline{U}_1 \\ \underline{U}_2 \end{bmatrix} \text{ leading to}$$

$$\underline{\mathbf{Y}}_{\Pi,\mathrm{T}} = \begin{bmatrix} \underline{Y}_{\mathrm{T}} & -\underline{Y}_{\mathrm{T}} \\ -\underline{Y}_{\mathrm{T}} & \underline{Y}_{\mathrm{T}} \end{bmatrix} - \begin{bmatrix} \underline{0} & (\underline{\vartheta} - 1)\underline{Y}_{\mathrm{T}} \\ -(\underline{\vartheta}^{*} + 1)\underline{Y}_{\mathrm{T}} & (\underline{\vartheta}^{*}\underline{\vartheta} + 1)\underline{Y}_{\mathrm{T}} \end{bmatrix} \tag{2.3}$$

Per unit system specialities

Reactances and resistances are referred to the base voltage and apparent power of the operational unit, such as the transformer. The power system simulation uses its own base voltage and base apparent power, enabling the use of one single calculation domain. This is done to simplify the calculation and to make the results easily comparable to each other. Hence, the reffered values have to be transformed from the equipment based values to the simulation based values. The relations and conversions are defined as follows.

$$\underline{Y}_{\mathrm{T}} = \frac{1}{r_{\mathrm{T}} + x_{\mathrm{T}} \cdot j} \cdot \frac{b_{\mathrm{T}} \cdot j}{2}$$

$$\underline{Y}_{\mathrm{T,\ sim}} = \underline{Y}_{\mathrm{T}} \cdot \frac{S_{\mathrm{n}}}{S_{\mathrm{n,\ sim}}} \tag{2.4}$$

$$\underline{U}_{\mathrm{whatever,\ sim}} = \underline{U}_{\mathrm{whatever}} \cdot \frac{S_{\mathrm{n}}}{S_{\mathrm{n,\ sim}}} \tag{2.5}$$

Displayed like in **??**, the characteristic of the operational unit is referred to the simulation base value. Here, the admittance of the transformer is multiplied with its own rated apparent power, then devided by the apparent power of the simulation system. Similar, the voltages are calculated via **??**. This specialities are considered in the tap changer modeling, thus further information is given in [**machowskiPowerSystemDynamics2020**] Appendix A.

Additionally to consider:

- D-q transformations (???),
- Frequency domains: reactances and inductances are dependent and can change with the base frequency,
- Torque and power relations.

2.1.3 Open-Source Power System Simulation tools

Some information about other open source python power system simulation tools, such as:

- Pandapower,
- TOPS,
- ... .

Build up like a scan (see Georg's thesis).

BUT: As well including the there used implementation of transformers mathematical background and complexity.

Table 2.1: Voltage instability types and different time frames with examples; after **[Quelle]**

No	Type	Cause of incident	Time frames
1	Long-term	Slowly use up of reactive reserves and no outage	Several minutes to several hours
2	Classical	Key outage leads to reactive power shortage	One to five minutes
3	Short-term	Induction motor stalling leads to reactive power shortage	Five to fifteen seconds

2.2 Voltage stability basics

2.2.1 Voltage stability definitions, classifications, and conditions

A Practical introduction to voltage stability assessment, methods and indices is given in the standard and extending literature of **rueda-torresEvaluationVoltageStability2024**, **danishVoltageStabilityElectric2015**, **cutsemVoltageStabilityElectric1998**.

Interesting to note/implement here: Basic classification, definitions, and the nature or conditions of voltage stability. Such as

- Short term vs. long term
- Static vs. dynamic
- Transmission driven vs. load driven vs. generation driven; stability/instability, and/or contributions
- Influence OLTC: Restoring voltage level, but not adding reactive capacities; hence adding risk of voltage collapses
- Load vs. transmission aspects
- Example mechanism: **Collapse effect of the nordic test system** [**vancutsemTestSystemsVoltage**
cutsemVoltageStabilityElectric1998]

2.2.2 Stability Indices

One easy idea for obtaining a stable operation is looking at the Jacobian Matrix. If this matrix is getting singular, the System will not remain in a stable operation. Singularity of matrices is checked by follwing two hypothesis tests:

$$\det(\mathbf{J}) = 0 \tag{2.6}$$

$$J \times J^{-1} \uparrow \tag{2.7}$$

The Jacobian Matrix is defined as:

$$\mathbf{J} = \begin{bmatrix} \Delta\mathbf{P} \\ \Delta\mathbf{Q} \end{bmatrix} = \left[\begin{array}{cc} \mathbf{H} & \mathbf{M'} \\ \hline \mathbf{N} & \mathbf{K'} \end{array}\right] \cdot \begin{bmatrix} \Delta\delta \\ \Delta V/V \end{bmatrix}$$

$$\left[\begin{array}{c} \Delta P_1 \\ \vdots \\ \Delta P_n \\ \hline \Delta Q_1 \\ \vdots \\ \Delta Q_n \end{array}\right] = \left[\begin{array}{cccccc} \frac{\partial P_1}{\partial \delta_1} & \cdots & \frac{\partial P_1}{\partial \delta_n} & V_1 \frac{\partial P_1}{\partial V_1} & \cdots & V_n \frac{\partial P_1}{\partial V_n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial P_n}{\partial \delta_1} & \cdots & \frac{\partial P_n}{\partial \delta_n} & V_1 \frac{\partial P_n}{\partial V_1} & \cdots & V_n \frac{\partial P_n}{\partial V_n} \\ \hline \frac{\partial Q_1}{\partial \delta_1} & \cdots & \frac{\partial Q_1}{\partial \delta_n} & V_1 \frac{\partial Q_1}{\partial V_1} & \cdots & V_n \frac{\partial Q_1}{\partial V_n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial Q_n}{\partial \delta_1} & \cdots & \frac{\partial Q_n}{\partial \delta_n} & V_1 \frac{\partial Q_n}{\partial V_1} & \cdots & V_n \frac{\partial Q_n}{\partial V_n} \end{array}\right] \cdot \left[\begin{array}{c} \Delta\delta_1 \\ \vdots \\ \Delta\delta_n \\ \hline \Delta V_1/V_1 \\ \vdots \\ \Delta V_n/V_n \end{array}\right] \tag{2.8}$$

Although this method seems easy to implement, there are some numerical problems realted to that. Checking if a Matrix is singular with numerical mathods, can only be realised as a probablilty expression. A result could be, that the determinant of the matrix is below a certain threshold. The algorithm would propose, that the matrix is probabilistic singular. [QUELLE] This problem leads to the necessity of applying other methods or indices for stability assessment. **danishVoltageStabilityElectric2015** is proposing a few other indices, that are based on the Jacobian Matrix, and shows comparitive characteristics between Jacobian Matrix and system variable based voltage stability indices. These Jacobian Matrix based indices are listed and further described in **??**, while the comparative characteristics are described in **??**.

2.2.3 Assessment methods

2.2.4 Analytical stability calculation of static power systems

2.3 Control engineering theory

2.3.1 Commonly used on-load tap changer control

A few basics are in the interest, understanding differences between real world beahavior, or possible ways of building up a **OLTC!** transformer control. This control theory difference can be limiting as well for the results and objectives compared to the actual possible control in the field.

The target voltage is typically set from the control room of the grid operator, coming from pre-calculated load flow analysis. This can be set hours before, or even day-ahead with the estimated loads of the grid. This value is set locally for each operating unit subsequently. The control is then operating locally and without further involvement of the grid operator. **[Quelle]**

Typically the used controller in the field is a discrete controller, which can change tap positions under load within a time frame of around few seconds. Practical tap steps are around $2\ \%$ of the overall transforming ratio. The control is set up with a dead band, to avoid unnecessary tap changes. It is necessecary to note here, that this control and its mathematical caracteristics contains logical elements, blocks, and delays, which cannot be translated in a typical control theory transmission function. This leads to the missing possibility to easily obtain mathematical stability for the control of the overall considered power system. **[Quelle]**

2.3.2 Dynamic voltage stability

Can I really express this as „Controller theory“?

2.3.3 Bifurcations and Chaos theory control

Is this necessary or already out of scope?

1. Fuzzy Control mechanisms,
2. Neural Networks,
3. Bifurcations.

3 Methodical Modeling

3.1 Transformer Equipment Modeling

Some literature and fundamentals about transformers, control, stability assessment, fast-switching modules, and analysis in Python.

Things to mention:

- First model: Admittance Matrix manipulations
- Second model: Current Injection Model

3.1.1 Implementing a Π-Representative Circuit with Variable Ratio

Mathematical Description and Definitions

Describe the Transformer circuit, the Π-model, simplification of impedances, relation to the specific side, ...

1. Show the admittance matrix deduction process for placing the impedances on one side
2. How to interprete $\underline{\vartheta}$; especially if complex and a mixture of longitudinal and angle ratio
   -> e.g. Asymmetrical shifter, vector groups, or only tapping on the current-voltage angle ϕ
3. Illustrate differences if the impedance relation is switching sides (Machowski vs. Kundur / Milano)
   -> refer the admittances to the lv or hv side: different definition of $\underline{\vartheta}$, different admittance matrix of the transformer
4. Dependency of connections? (which side, step-up, step-down transformer?)

Additional Algorithmics

- How to automatically determine switching direction?
  -> switchin direction dependent on what? (load-flow direction?)
- Controller set points: also dependent on load flow?
- How can I change the transformer control setpoints to be load flow dependent?
- How can I ensure, utilization of the transformer is not $> S_\mathrm{n}$?

3.1.2 Tap Changer Control Modeling

This is the description of the ideas, development, and implementation of a OLTC control scheme.

Discrete Control Loop

- Describe implementation
- Describe benefits / drawbacks
- Control scheme
- Switching logic and behavior (voltage tracking)

This control method represents the currently most used and thus representative control scheme for **OLTC!**s. With the mechanic nature of the switching mechanism, the control look can only access discrete ratios within time frames of around a few seconds. Such a discrete control loop is described by **milanoHybridControlModel2011**, **milanoPowerSystemModelling2010**. A scheme of this control loop is shown in **??**.

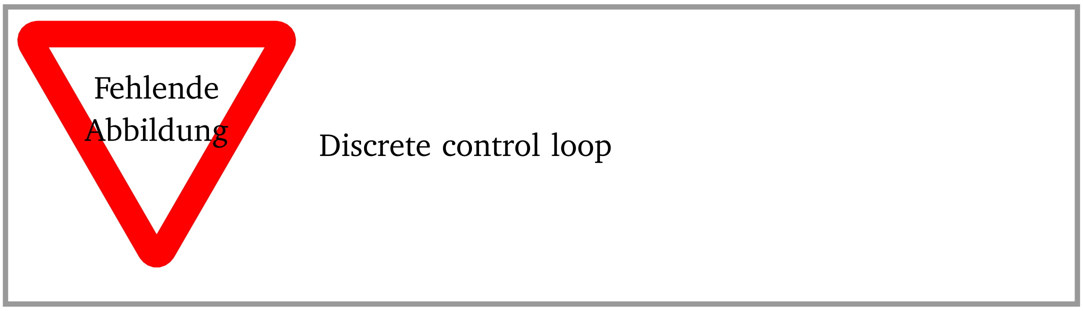

Figure 3.1: Discrete control loop of an **OLTC!**; scheme based on **milanoHybridControlModel2011**

This control loop type is beneficial due to its accurate representability of current **OLTC!** abilities. It gains access to assess stability within simulation environments, as analytical methods are not suited.

A negative aspect of a discrete control loop is the missing opportunity of generating a transfer function. This blocks the stability assessment with standard control engineering methods. Further, popular analysis methods like eigenvalue analysis is not possible, due to the lack of possibility to form derivatives.

The structure of the implementation is illustrated in the block diagram of **??**. The controller is actively chainging the algebraic funtions of the simulation environment, therefore it is quasi dynamic. The controller output logic is called, when updating the admittance matrix of the transformer. Additionally, the differential functions of the connected simple controllers, like integrators, PT1-blocks, etc., are called by the solver and are thus part of the differential equations. The logic determines the physical interpretation of the **OLTC!**, and therefore

1. If the OLTC has to switch,
2. When the switching operation is finished, and
3. What the current, or in case after a switching the new, tap ratio is.

It is important to note, that this structure relies on the calculation of the dynamic admittance matrix on each time step.

Implementation structure of the discrete OLTC controller

Figure 3.2: Implementation structure of the discrete OLTC controller

The output of the controller is based on the following logic.

Continous Control Loop

Control Schemes for the Fast Switching module

- Describe implementation

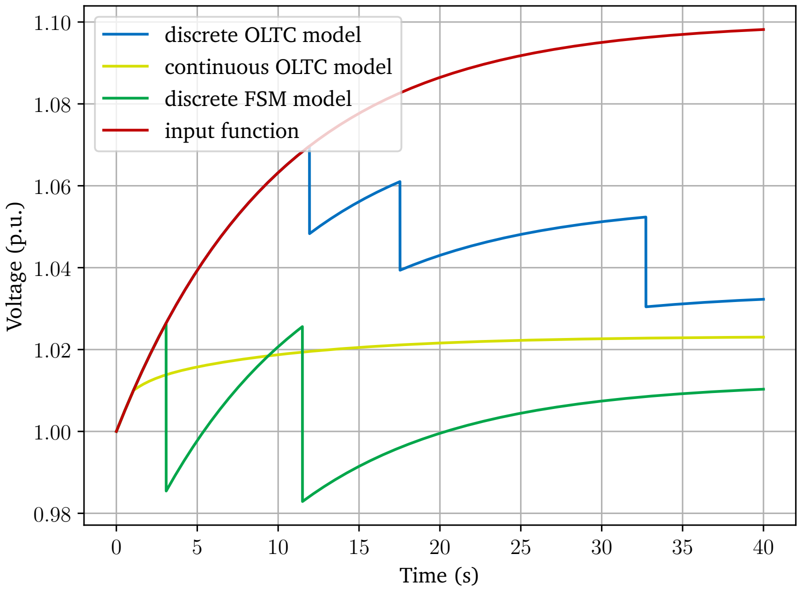

Figure 3.3: Characterization of the OLTC control loop; the input function simulates the to be regulated voltage, the output functions are characterized by $o(t) = i(t) \cdot \underline{\vartheta}_{\text{trafo}}$

- Describe benefits / drawbacks
- Control scheme
- Switching logic and behavior (voltage tracking)

Describe the operational logic and structure of the **FSM!** (**FSM!**) first.

A control logic for a so called **FSM!** has been presented from **burlakinEnhancedVoltageControl202** and illustrated in **??**.

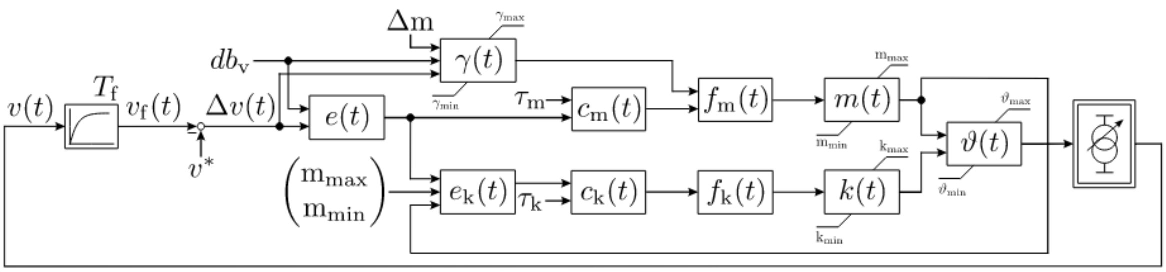

Figure 3.4: Control loop of a **FSM!**; scheme based on **burlakinEnhancedVoltageControl2024**

However, the implementation logic in Python is slightly differing from the presented scheme in [**burlakinEnhancedVoltageControl2024**], simply for not overcomplication of the code and therefeore debugging. The implementation is similar to the afore discussed one of a standard **OLTC!** controller.

Discrete Control Loop as most Representative

A continous control loop for a **FSM!** is presented within **burlakinEnhancedVoltageControl2024**, **burlakinEnhancingVariableShunt2024**. Similar to the solely **OLTC!** loop, it represents the real behavior best, but is obstructive for stability assessments. The scheme of the logic is shown in **??**.

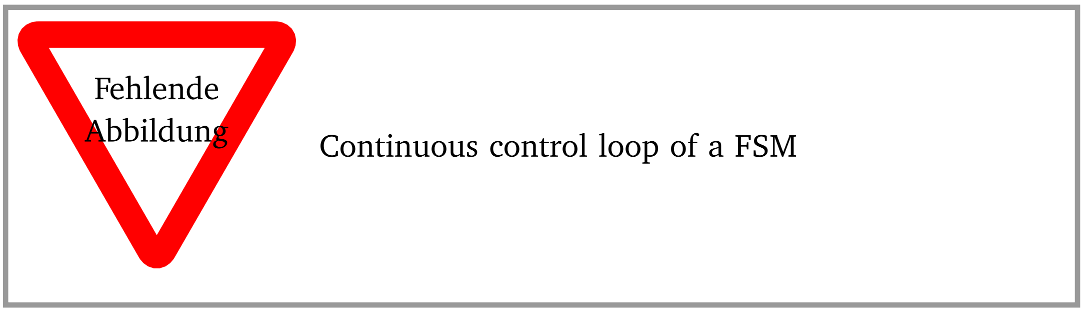

Figure 3.5: Continuous control loop of a **FSM!**; scheme based on **burlakinEnhancedVoltageControl2024**

Continuous Control Loop for best Stability Assessment

3.1.3 Experimental: Extended Ideas and Improvements

Operational Oriented FSM Control

Alternative Tap Skipping Logic

Varying the Voltage Setpoint and Target Calculation

Here, another idea of control target creation shall be mentioned. Instead of a fixed bus voltage reference, the difference of both bus voltages is considered. Further, the sign of that difference is used to determine the direction of the tap change.

3.2 Supplementary Modeling and Advancements

As the python framework is currently missing some representations of components, this chapter aims to describe the implementation of those. Mainly focussing on source and load models, as the later considered test, benchmark, and use case networks require alternative behaviors.

3.2.1 ZIP Load Model

Why important?

Mostly, a polynomial load model is used. It is called ZIP-model, as there are individual contributions to constant impedance $\underline{Z}$, constant current $\underline{I}$, and constant power P, or respectively Q, are considered. The model is described by **IEEEGuideLoad2022**. Either two ways of mathematical description are considered valid, dependent on the allowed influence of the frequency deviation. The use of periodized phasor representation, typical for a **RMS!** simulation, is missing or often neglecting this frequency information. Therefore the set of **??** and **??** is considered sufficient and implemented in the Python framework.

$$P = P_n \cdot \left[p_1 \left(\frac{U}{U_n} \right)^2 + p_2 \left(\frac{U}{U_n} \right) + p_3 \right] \tag{3.1}$$

$$Q = Q_n \cdot \left[q_1 \left(\frac{U}{U_n} \right)^2 + q_2 \left(\frac{U}{U_n} \right) + q_3 \right] \tag{3.2}$$

$$\text{with} \quad p_i \in [0,1] \quad \text{and} \quad q_i \in [0,1]$$

Characteristics?

How does it look like in the simulation environment?

$$P = P_n \cdot \left[p_1 \left(\frac{U}{U_n} \right)^2 + p_2 \left(\frac{U}{U_n} \right) + p_3 \right] \cdot (1 + k_{pf} \Delta f) \tag{3.3}$$

$$Q = Q_n \cdot \left[q_1 \left(\frac{U}{U_n} \right)^2 + q_2 \left(\frac{U}{U_n} \right) + q_3 \right] \cdot (1 + k_{qf} \Delta f) \tag{3.4}$$

$$\text{with} \quad \sum_{i=1}^{3} p_i = 1 \quad \text{and} \quad \sum_{i=1}^{3} q_i = 1$$

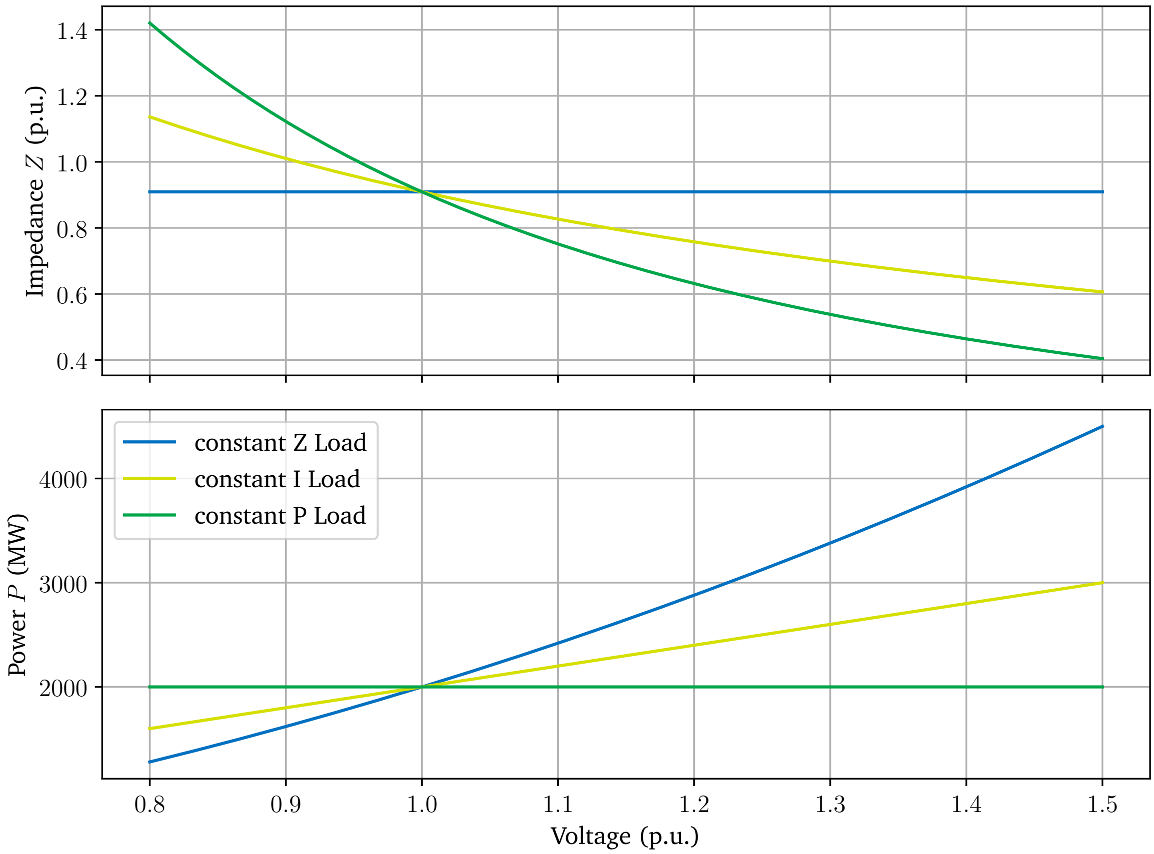

Figure 3.6: Characterization of the ZIP load model; with (upper) the result of the impedances dependent on the voltage at the connected bus, and (lower) the resulting power consumption of the different models, representative only the real power P

Although the subset of **??** and **??** would consider a relation to the frequency at the given time, it is not used. The use of periodized phasor representation, typical for a **RMS!** simulation, is missing this information in the framework diffpssi. Therefore the set of **??** and **??** is implemented in the Python framework. It has to be mentioned, because the comparison tool PowerFactory is using this load model, though in some parts of the simulation results are showing a slightly bigger error.

3.2.2 Induction Machine Models

As one of the most important loads to consider, especially for many load driven instability mechanisms, the **IM!** (**IM!**) is a crucial component, **[Quelle]**

Just briefly:

- Why is it crucial?

- How do the instability mechanisms work and look like?
- What are the different types of **IM!**s modeling (complete and dynamic, static, ...)

Three main ways of **IM!** modeling are relevant to mention in this section:

1. Static model as introduced in **IEEEGuideLoad2022**,
2. a dynamic 'fixed-speed' **IM!** model, and
3. a doubly fed **IM!** model.

The last ones are mentioned and further described in **machowskiPowerSystemDynamics2020**. Least model requires very detailed information, and shall be suitable for **SMIB!** models for machine behavior studies or similar. The second model is suitable for network analysis and machine behaviors. The first model applies for high perception of **IM!**s in total loading of the network. As referencing to **IEEEGuideLoad2022**, is is similar implemented as the before mentioned ZIP load model, considering characteristic equations for its real power P and reactive power Q. Both models shall be described in the following section.

Static Model of Induction Machines

For this operational unit type is a detailed dynamic modeling possible. With some considerations, it can be sufficient, modeling this equipment just with the The model is described by **IEEEGuideLoad2022** as formulated in following set of equations.

$$P = \left(R_{\mathrm{s}} + \frac{R_{\mathrm{r}}}{s}\right) \cdot \frac{U^2}{\left(R_{\mathrm{s}} + \frac{R_{\mathrm{r}}}{S}\right)^2 + (X_{\gamma\mathrm{s}} + X_{\gamma\mathrm{r}})^2} \tag{3.5}$$

$$Q = (X_{\gamma\mathrm{s}} + X_{\gamma\mathrm{r}}) \cdot \frac{U^2}{\left(R_{\mathrm{s}} + \frac{R_{\mathrm{r}}}{S}\right)^2 + (X_{\gamma\mathrm{s}} + X_{\gamma\mathrm{r}})^2} + \frac{U^2}{X_{\mathrm{s}}} \tag{3.6}$$

[MK]Is here really a difference between the two s in the equations?

Briefly describe the implementation.

Dynamic 'fixed-speed' Induction Machine model'

From ChatGPT:

The dynamic model of **IM!**s is essential for accurately representing their behavior under various operating conditions. This model includes the differential equations that describe the machine's electrical and mechanical dynamics. The equations are typically derived from the machine's equivalent circuit and can be expressed in the d-q reference frame.

The dynamic model can be represented by the following set of equations:

$$\frac{\mathrm{d}\psi_{\mathrm{d}}}{dt} = v_{\mathrm{d}} - R_{\mathrm{s}} i_{\mathrm{d}} + \omega \psi_{\mathrm{q}} \tag{3.7}$$

$$\frac{\mathrm{d}\psi_{\mathrm{q}}}{dt} = v_{\mathrm{q}} - R_{\mathrm{s}} i_{\mathrm{q}} - \omega \psi_{\mathrm{d}} \tag{3.8}$$

$$\frac{\mathrm{d}\omega}{dt} = \frac{1}{J}(T_m - T_e - B\omega) \tag{3.9}$$

where:

- $\psi_{\mathrm{d}}, \psi_{\mathrm{q}}$ are the d-q axis flux linkages
- $v_{\mathrm{d}}, v_{\mathrm{q}}$ are the d-q axis voltages
- $i_{\mathrm{d}}, i_{\mathrm{q}}$ are the d-q axis currents
- R_{s} is the stator resistance
- ω is the rotor angular velocity
- T_{m} is the mechanical torque
- T_{e} is the electromagnetic torque
- J is the moment of inertia
- B is the damping coefficient

The electromagnetic torque T_e can be calculated as:

$$T_e = \frac{3}{2} p(\psi_d i_q - \psi_q i_d) \tag{3.10}$$

where p is the number of pole pairs.

This dynamic model allows for the simulation of the **IM!**s transient response to changes in voltage, frequency, and load conditions. It is particularly useful for studying stability and control strategies in power systems.

Briefly describe the implementation.

3.3 Application of Voltage Stability

3.3.1 Influences of other device characteristics

Just look on other mutual influences in the power system (simulation), such as:

- Load characteristics and types of modeling
- Maximum thermal currents of cables and operating components
- Asynchronous machines (or called „induction motors“?)

3.3.2 Observing the current state of the system

Static and Dynamic Indices

- Which indices can be implemented?
- Which make sense?
- Implementation and calculation of them?

Stability Monitoring

- Index combination and „traffic light“m̃onitoring
- Restauration options and opportunities
- Local mapping
- Weak point identification

4 Verification setup and results

4.1 Representative Electrical Networks

The following section shall introduce the used power systems in the simulation with the Python framework, considering verification, and also extension meaning the performed case studies in **??**. The models are chosen to represent different network sizes and complexities, thus allowing the objective of graded interaction levels of the developed (transformer) model. The models are based on the work of **machowskiPowerSystemDynamics2020**, **kundurPowerSystemStability2022**, **IEEEGuideLoad2022**, and **vancutsemTestSystemsVoltage2020**.

Single Machine Infinite Bus (SMIB) Model

One very popular and thus powerful electrical network for the verification of power system stability is the **SMIB!** model. It is a compact and simplified model of a power system, allowing easy analytical calculation, verification and development. Mutual influences are comparably simple to understand and calculate, as the infinite bus bus is acting as a fixed grid connection point with a large adjoining grid. The generator is connected to the bus bar via a transmission line and a transformer. The model was largely discussed by **kundurPowerSystemStability2022**, and is shown in Figure **??**. The generator and the **IBB!** are represented by synchronous machines, developed and discussed by **kordowichPhysicsInformedMachine2023**. The specific model details are included in **??**, additionally the simulation setup for verification is described in **??**.

Figure 4.1: **SMIB!** (**SMIB!**) model for verification and validation of the Python framework; own figure after [**machowskiPowerSystemDynamics2020**, **kundurPowerSystemStability2022**]

Table 4.1: Simulation Setup for validation of the Π-modeled transformer; considering a transforming ratio $\underline{\vartheta} \neq 1$ and $\underline{\vartheta} \in \mathbb{C}$

Parameter	**Value**
Generator inertia H	3.5 s
Generator damping D	0.1 p.u.
Generator resistance R	0.01 p.u.
Generator reactance X	0.1 p.u.
Transformer resistance R	0.01 p.u.
Transformer reactance X	0.1 p.u.
Transmission line resistance R	0.01 p.u.
Transmission line reactance X	0.1 p.u.

Further, this model shall be slightly modified according to **??**. A load is added at the secondary bus of the transformer, the rest of the system is kept. **??** already contains this modification.

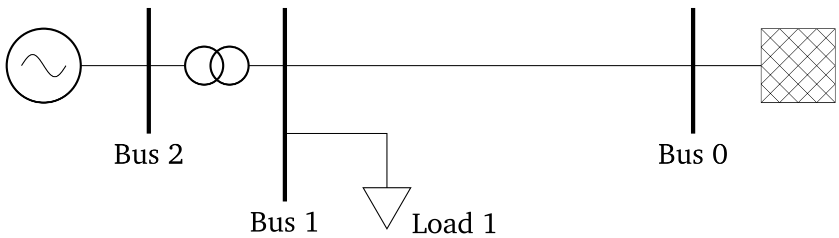

Figure 4.2: Modified **SMIB!** (**SMIB!**) model with additional load

Simple Single Machine Load Model

Following model is often recommended **[Quelle]** for easy voltage control studies, in explicit for **OLTC!**s. Similar to the **SMIB!** model, it consists from one synchronous generator, busses, and lines in a single branch. The **IBB!** is thus removed and changed to a load. This two element type o configuration allows for an easy analytical calculation of voltage stability and control. Although this thesis is focussing on **OLTC!** transformers, the model is extended with one in between. A single line representation is depicted in **??**.

Further details about its configuration and simulation setup are included in **??**. It should be noted, that simple load models are not useful for simulation of this example

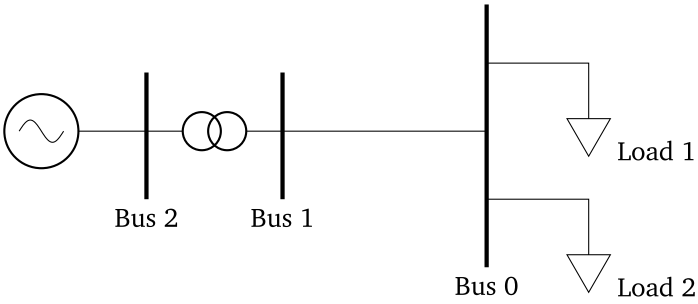

Figure 4.3: Single line representation of a simple single machine load model; own illustration with characterstics from **[Quelle]**

network. Usually constant Z models are used as loads, therefore simulation results can be misleading and not showing desired effects or voltage instability mechanisms **[Quelle]** . The simulation framework is extended with XX types of load models, to satisfy the requirements of the single machine load model, and a connected stability assessment.

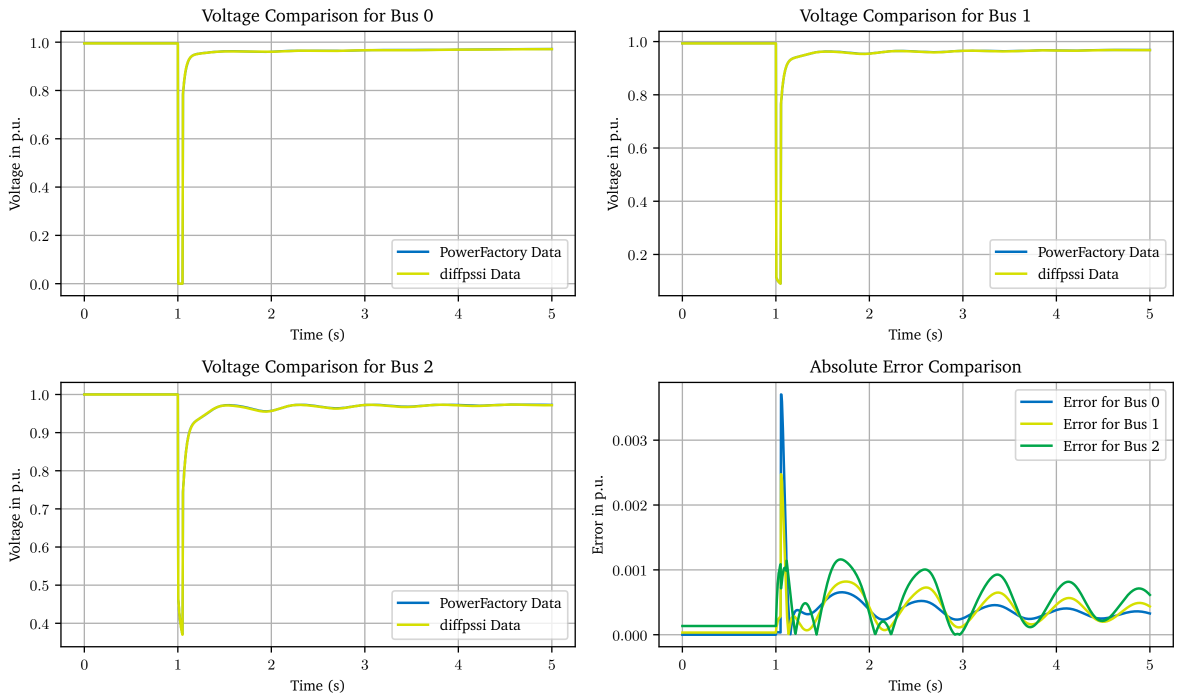

Figure 4.4: Comparison of the Π-modeled transformer in the **SMIB!** model between PowerFactory and the Python framework

IEEE nine-bus system

Nordic test system

4.2 Validation Steps

4.2.1 Validation of the Π-modeled Transformer with Variable Tap Position

4.2.2 Validation of the ZIP-Load Model

4.2.3 Validation of the OLTC control schemes

Standard Discrete OLTC Control

Fast Switching OLTC Control

4.2.4 Validation of the Induction Machine Model

Place results here, looking at: off nominal tap ratio, and with off nominal phase shifting (e.g. $110°$)

4.3 Discussion of Model Limitations and Improvements

5 Case study

In the interest of investigation / the Case Study are:

- Influence of switching times on stability margin/begin of destabilization,
- Influence of max. ratio change per switching event, and
- Influence on different test systems (destabilization mechanisms).

5.1 Scenario setting

Does it make sense to structure like that? (Scenarios - Simulation - Results)

Or is it a better idea thinking in terms of specific „use cases“ as sections:

- What happens under strong grid conditions? -> Section: Strong grid condition behavior
- What happens under weak grid conditions? -> Section: Weak grid condition behavior
- Strongly interconnected grids
- Widely extended linear string grids
- Section: Use case of Wind farm integration
- Influence on transient stability: SMIB model with and without OLTC

Influence of FSM on Machines and their stability criterions

Thinking of Rotor Angle Stability, maybe considered by an EAC implementation? What does the fast Switching, esp. at up to 8% of the nominal voltage, do to the machines?

Novel Control Strategy FSM

Thinking of fast and slow voltage gradients: fast gradients are compensated by the FSM, slow gradients are compensated by the OLTC. Therefore optimal utilisation of injected damping moment of the FSM.

Also thinking of different presets of the OLTC and FSM, which are tried to keep constant. Different grid operators can utilize for typical grid conditions of over- or undervoltage at **PCC!**.

Following contains:

- Implemetation of different logic
- Testing of presets and switchin logic
- Damping moment beneficial?

Possible Extension for Power Flow congruent Control

Extension in the Control Algorithm to decide which Bus has to be regulated, to avoid contrairy actions of the OLTC and FSM against the power flow. Therefore not decrease of stability, but increase. Possible Application: Grid coupling Transformers, Battery Storage assisted Virtual powerplants, etc.

For this thinking maybe another control strategy is relevant:

- No setpoint from a load flow day-ahead or similar time frame; but rather current load and bus voltages are considered
- Not the deviation of one transformer bus voltage from a setpoint, but the deviation between the two bus voltages is relevant
- Maybe the absolut deviation to the optimal bus voltages at the current load situation is relevant

Big general problem: In which direction does the OLTC / the FSM have to swith? In some cases, the direction is not correct, in some it is correct.

5.2 Simulation

5.3 Results

6 Discussion of the results

7 Summary and outlook

Some conclusion.

Some outlook and nice blibla.

Acronyms

Symbols

δ	$^\circ$ / deg	power angle (or power angle difference)
$\Delta\omega$	$\frac{1}{\mathrm{s}}$	change of rotor angular speed
$\underline{\theta}$	-	transformer ratio; complex if phase shifting
A	-	acceleration or deceleration area
$\underline{E}$	V	voltage of **SG!** or **IBB!**
H_{gen}	s	inertia constant of a **SG!** (**SG!**)
$\underline{I}$	A	current
P	W	effective power; electrical or mechanical
Q	var	reactive power
R	Ω	ohmic resistance
$\underline{S}$	VA	apparent power
$\underline{V}$	V	voltage
$\underline{X}$	Ω	reactance
$\underline{Y}$	$\frac{1}{\Omega}$ / S	admittance
$\underline{Z}$	Ω	impedance

The different symbols are used with different indices, these are semantic and explained in the surrounding context. Following notation is commonly used for mathematical and physical symbols:

- Phasors or complex quantities are underlined (e.g. $\underline{I}$)
- Arrows on top mark a spatial vector (e.g. $\overrightarrow{F}$)
- Boldface denotes matrices or vectors (e.g. **F**)
- Roman typed symbols are units (e.g. s)
- Lower case symbols denote instantaneous values (e.g. $\underline{i}$)
- Upper case symbols denote **RMS!** or peak values (e.g. $\underline{I}$)
- Subscripts relating to physical quantities or numerical variables are written italic (e.g. $\underline{I}_1$)

In the simulations and calculations the per unit system ($\mathrm{p.u.}$) is preferred, thus normalizing all values with a base value. Where necessary, absolute units are added to indicate the explicit use of the normal unit system. For more information about this per-unit system please refer to **machowskiPowerSystemDynamics2020**, specifically Appendix A.1 provides a detailed description and explanation.

List of Figures

List of Tables

List of Listings

Appendix

A Fundamentals

A.1 Description of the Power System Simulation process

In this appendix section, the general process of power system simulation is described. As this thesis is aiming to understand voltage stability and processes in longer periods of time, these explanations apply to pointer-based simulations, called RMS simulations. Meaning that the considered effects are slower electromechanical nature instead of faster electromagnetic ones. The in this thesis used Python framework „*diffpssi*“ is based on this type of simulation, and due to its open-source based nature traceable.

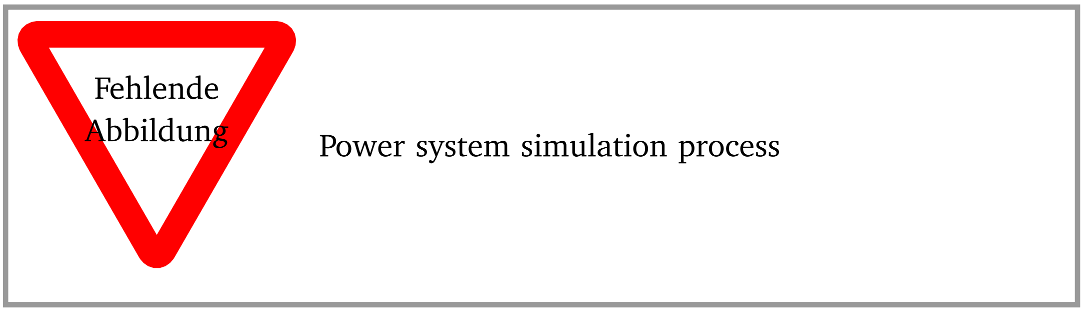

Figure A.1: Power system simulation process; own illustration

Really basic: (?)

- Phasor description
- Symmetricak Components
- RMS vs EMT simulation (-> meaning one cannot simulate other faults than 3ph w/o ground)

Less basic and more advanced:

- rountines in the framework
- two types: Algebraic and Differential equations have to be solved at each time step -> What is which? Which operational equipment is typically described with which type of equation?

A.2 Jacobian based voltage stability criterions

danishVoltageStabilityElectric2015 is showing, describing, and referencing some voltage stability indices based on the Jacobian matrix. The following table is a collection of these indices.

A.3 Comparison of System based and Jacobian based indices

Table A.1: Jacobian based voltage stability criterions; after **danishVoltageStabilityElectric2015**

Index	**Abbreviation**	**Calculation**	**Stability Threshold**	**Reference**
Tangent Vector Index	**TVI!**	$\mathrm{TVI}_i = \left\lvert \frac{\mathrm{d}V_i}{\mathrm{d}\lambda} \right\rvert^{-1}$	depending on load increase	
Test Function		$t_{cc} = \left\lvert e_c^T \cdot \mathbf{J} \times \mathbf{J}_{cc}^{-1} \cdot e_c \right\rvert$	details are given in reference	
	i	$i = \frac{1}{i_0} \cdot \sigma_{\max} \cdot \left(\frac{\mathrm{d}\sigma_{\max}}{\mathrm{d}\lambda_{\mathrm{total}}}\right)^{-1}$	$i > 0$	
Minimum Eigenvalue		$\Delta V = \sum_i \frac{\xi_i \eta_i}{\lambda_i} \Delta Q$	all eigenvalues should be positive	
Minimum Singular Value		$\begin{bmatrix} \Delta\vartheta \\ \Delta V \end{bmatrix} = \mathbf{V} \sum^{-} 1 \mathbf{U}^T \begin{bmatrix} \Delta F \\ \Delta G \end{bmatrix}$	details are given in reference	
Predicting Voltage Collapse		$\frac{V}{V_0}$	the smallest index value	
Impedance Ratio		$\frac{Z_i i}{Z_i}$	$\frac{Z_i i}{Z_i} \leq 1$	