

Semesterarbeit

Applikation zur Berechnung der akustischen Geräuschemissionen von Hochspannungsfreileitungen



Felix Jauch

Betreuer: Dr. Ulrich Straumann

Prof. Dr. K. Fröhlich

14. Februar 2010

Abstract

Unter schlechten Wetterbedingungen wie Regen, Schnee oder Nebel kommt es an Hochspannungsfreileitungen zu akustischen Geräuschemissionen. Zur Vorhersage solcher Emissionen in Form A-gewichteter Schallpegel wird eine Java-Anwendung entwickelt, welche beliebige Mastgeometrien mit Dreiphasenwechselstromsystemen 50 Hz, Bahnstromsystemen $16\frac{2}{3}$ Hz und Gleichstromsystemen berücksichtigt. Die Berechnung der akustischen Geräuschemissionen erfolgt nach dem Formelwerk des Electric Power Research Institute (EPRI). Die dazu erforderlichen elektrischen Randfeldstärken der Teilleiter werden mittels Ersatzladungsverfahren ermittelt.

Studienarbeit «Programmieren einer Anwendung zur Berechnung der akustischen Geräuschemissionen von Hochspannungsfreileitungen»

Einleitung

Insbesondere benetzte Hochspannungsfreileitungen erzeugen ein hochfrequentes, prasselndes Geräusch und ein tieffrequenten, tonalen Brummtönen. Beides kann von Anwohnern als störend empfunden werden. Die beiden Komponenten der Emission haben zwar unterschiedliche Erzeugungsmechanismen, die Schlüsselrolle spielen jedoch in beiden Fällen die Wassertropfen auf den Leiterseilen, die eine Vergrößerung der elektrischen Feldstärken in ihrer Umgebung verursachen, was zu Entladungsaktivität führen kann. Die hochfrequente Emission war speziell ab den 70-er Jahren des vergangenen Jahrhunderts Gegenstand intensiver Forschungstätigkeit, als erste Überlegungen zur Steigerung der Betriebsspannung der Leitungen bis über $1'000kV$ im Gange waren. Die tonale Emission blieb dabei weitgehend unbeachtet. Im Rahmen des Projekts CONOR wurde der Mechanismus der zu dieser Emission führt verstanden und eine Methode zur Berechnung der emittierten Pegel entwickelt. Im Projekt Conor sind zahlreiche Industrievertreter beteiligt, die an einem Werkzeug zur Berechnung der akustischen Geräusche sehr interessiert sind. Eine Möglichkeit hierzu ist die Erstellung einer entsprechenden Anwendung.

Ziele

- Anwendung zur Berechnung der akustischen Emission von beliebigen Hochspannungsdrehstromfreileitungen bei Regen basierend auf den bestehenden Berechnungsmethoden.
- Implementierung von Möglichkeiten zur Weiterverarbeitung der berechneten Pegel durch graphische Darstellung und Exportfunktionen.
- Zusätzliche Darstellung der berechneten elektrischen Randfeldstärken.

Aufgaben

Implementierung des Berechnungsalgorithmus sowie Schaffen einer graphischen Benutzeroberfläche, über die der Benutzer die Leiterparameter eingeben und die resultierenden Rechenergebnisse weiterverarbeiten kann.

Start der Arbeit 15. September 2009
Ende der Arbeit 18. Dezember 2009
Bearbeiter Felix Jauch
Betreuer Dr. Ulrich Straumann

Inhaltsverzeichnis

1	Einleitung	1
1.1	Akustische Geräuschemissionen von Hochspannungsfreileitungen	1
1.2	Mass zur Quantifizierung akustischer Geräuschemissionen	2
1.3	Grundstruktur der Applikation <i>HVLNoise</i>	2
2	Eingabe und Speicherung der Leitungsparameter	5
2.1	Benutzeroberfläche zur Eingabe der Leitungsparameter	5
2.2	Hierarchische Datenstruktur zur Speicherung der Leitungsparameter	7
2.2.1	Hochspannungsfreileitung (Klasse <i>HighVoltageLine</i>)	8
2.2.2	Elektrische Systeme (Klasse <i>ElectricalSystem</i>)	11
2.2.3	Leiter (Klasse <i>Conductor</i>)	13
2.2.4	Teilleiter (Klasse <i>Subconductor</i>)	15
2.3	Schlussfolgerungen	16
3	Berechnung der elektrischen Randfeldstärken mittels Ersatzladungsverfahren	17
3.1	Mathematische Formulierung des Ersatzladungsverfahrens	17
3.2	Platzierung der Konturpunkte und Ersatzladungen	19
3.3	Herleitung der Potentialkoeffizienten	20
3.4	Algorithmus zur Berechnung der elektrischen Randfeldstärken	22
3.4.1	Definition der Potentialverläufe	22
3.4.2	Berechnung der Ersatzladungen	24
3.4.3	Berechnung der RMS-Randfeldstärken	26
3.5	Genauigkeit	30
3.6	Vergleichsrechnung mit <i>Comsol</i>	31
3.7	Schlussfolgerungen	33
4	Berechnung der akustischen Geräuschemissionen	35
4.1	Berechnung der Schallpegel	35
4.1.1	Schallpegel L_5	35
4.1.2	Schallpegel L_{50}	37
4.1.3	Energieäquivalenter Schallpegel L_{eq}	38
4.1.4	Berücksichtigung der Regenrate	38
4.2	Benutzeroberfläche	41
4.3	Implementierung der Berechnung der akustischen Geräuschemissionen	41
4.4	Schlussfolgerungen	44
	Abbildungsverzeichnis	44
	Tabellenverzeichnis	45
	Literaturverzeichnis	47

1 Einleitung

Unter schlechten Wetterbedingungen, insbesondere bei Regen, Schnee oder Nebel, kommt es an Hochspannungsfreileitungen zu akustischen Geräuschemissionen, welche von Anwohnern als störend empfunden werden können. Für die Betreiber von Hochspannungsnetzen ist es daher unerlässlich, die akustischen Emissionen bestehender aber auch geplanter Hochspannungsfreileitungen abzuschätzen. Durch die Programmierung eines entsprechenden Werkzeugs soll die Vorhersage akustischer Geräuschemissionen in Form A-gewichteter Schallpegel für beliebige Leitungsgeometrien möglich sein.

1.1 Akustische Geräuschemissionen von Hochspannungsfreileitungen

Werden an den Leiterseilen von Hochspannungsfreileitungen bestimmte elektrische Feldstärken überschritten, so führt dies zur Ionisierung der umgebenden Luft und damit zu Koronaentladungen. Unter nassen Wetterbedingungen sammeln sich Wassertropfen auf der Oberfläche der Leiter, wodurch das elektrische Feld in naher Umgebung verstärkt und somit die Entladungsaktivität begünstigt wird.

Die akustischen Geräuschemissionen der Leiterseile bestehen aus einem hochfrequenten, prasselnden Geräusch sowie einem tieffrequenten, tonalen Brummtönen [2]. Das Geräusch der breitbandigen, hochfrequenten Komponente wird durch Koronaentladungen hervorgerufen. Die einzelnen Entladungen bilden dabei unkorrelierte Schallquellen. Die tonale Komponente doppelter Netzfrequenz entsteht durch elastische Stöße von durch Korona gebildeten und im elektrischen Wechselfeld beschleunigten Ionen mit hauptsächlich neutralen Teilchen [4]. Im Folgenden wird nur die hochfrequente, breitbandige Komponente betrachtet, da der tonale Brummtönen im A-gewichteten Schallpegel vernachlässigbar ist [2].

Zu Beginn einer Regenperiode, wenn die Leiterseile nicht durchgängig nass sind, wird ein stark schwankender Geräuschpegel wahrgenommen, je nach Regenintensität [2]. Sind die Leiter schließlich vollständig mit Wasser überzogen, nimmt die Fluktuation des emittierten Pegels ab, da auftreffende Regentropfen die zur Erde herabfallenden Tropfen ersetzen. Die höchste akustische Emission wird bei Sättigung der Leiterseile mit Wasser erreicht, wie im Fall von starkem Regen.

Bei starkem und länger anhaltendem Nebel können sich ebenfalls Wassertropfen auf den Leiterseilen bilden, die zu Geräuschemissionen führen. Bei Schnee spielt die Temperatur eine wesentliche Rolle [2]. Nahe $0\text{ }^{\circ}\text{C}$ ist mit Nassschnee und damit einer Emission vergleichbar bei starkem Regen zu rechnen. Deutlich unter $0\text{ }^{\circ}\text{C}$ ist der Schnee eher trocken, womit der Geräuschpegel deutlich schwächer wird.

1 Einleitung

Durch das häufigere Auftreten und den grossen Einfluss von Regen auf die akustischen Geräuschemissionen von Hochspannungsfreileitungen werden Leitungsgeometrien hinsichtlich der emittierten Geräuschpegel meistens unter regnerischen Wetterbedingungen beurteilt. Die Applikation zur Vorhersage solcher Pegel soll sich daher auf die Berechnung bei Regen beschränken.

1.2 Mass zur Quantifizierung akustischer Geräuschemissionen

Geräuschemissionen werden durch kleinste zeitabhängige Druckschwankungen hervorgerufen, welche sich dem atmosphärischen Druck überlagern. Als Mass akustischer Emissionen dient das Verhältnis des RMS-Werts der kleinen Druckschwankungen zu einem Referenzdruck. Dies führt auf die Definition des Schalldruckpegels gemäss Gleichung (1.1).

$$L_p = 20 \cdot \log_{10} \left(\frac{p_{rms}}{p_0} \right) \text{ dB} = 10 \cdot \log_{10} \left(\frac{p_{rms}^2}{p_0^2} \right) \text{ dB} \quad (1.1)$$

Als Referenzdruck wird im Folgenden, wie üblich, $p_0 = 20 \mu\text{Pa}$ verwendet. Die Angabe des Schalldruckpegels erfolgt also immer in Dezibel über $20 \mu\text{Pa}$.

Das menschliche Gehör nimmt vor allem Geräusche in Frequenzbereichen wahr, wo Sprachinformationen übermittelt werden. Es ist daher sinnvoll, bei Messungen von Geräuschemissionen nur hörbare Frequenzen zu berücksichtigen und entsprechend der Frequenzempfindlichkeit zu bewerten. Dies führt auf eine Gewichtung der bei verschiedenen Frequenzen gemessenen Schalldruckpegel. Für die Messung von akustischen Emissionen von Hochspannungsfreileitungen werden Frequenzfilter mit der A-Gewichtung eingesetzt [2]. Die Angabe der Geräusche erfolgt als Schallpegel, welcher in dB(A) angegeben wird.

Da Koronaentladungen zu zeitvariablen Schalldruckpegeln führen, werden statistische Schallpegel eingeführt, die sogenannten *L-Level*. Der Level L_{10} besagt zum Beispiel, dass der Schallpegel 10% der Zeit überschritten wird. Im Folgenden werden die Level L_5 und L_{50} relevant sein.

Ebenfalls gebräuchlich ist die Angabe eines äquivalenten Dauerschallpegels L_{eq} . Dieser stellt die Mittelung der Energie eines meist A-gewichteten Schallpegels einer zeitveränderlichen Schallquelle über eine bestimmte Periode dar. Ein dauerhaftes Geräusch mit gleichem Schallpegel wie der Äquivalenzpegel L_{eq} beinhaltet dieselbe Energie wie die des zeitveränderlichen. Auch diese Beschreibung einer akustischen Geräuschemission wird im Folgenden Anwendung finden.

1.3 Grundstruktur der Applikation *HVLNoise*

Um die Applikation zur Berechnung der akustischen Geräuschemissionen einem breiten Publikum zugänglich zu machen, insbesondere den im Projekt CONOR (Corona Noise Reduction) der ETH Zürich vertretenen Partner, wird die Programmiersprache Java gewählt. Dies

erlaubt die Benutzung der Anwendung unter verschiedensten Betriebssystemen ohne zusätzliches Kompilieren für die entsprechende Plattform. Die Voraussetzung ist lediglich ein Java-Interpreter für das jeweilige System.

Für die Ausführung von mathematischen Operationen auf Matrizen wird das Java-Paket *JAMA*¹, für die grafische Visualisierung von Rechenergebnissen das Paket *JFreeChart*² verwendet. Durch die Einbindung der genannten Pakete wird die Programmierung mit Matrizen sowie das Plotten von Daten erheblich vereinfacht.

Grundsätzlich wird die Applikation in drei Hauptklassen aufgeteilt. Zunächst werden die durch den Benutzer eingegebenen Leitungsparameter gespeichert. Dies geschieht über die Klasse *HighVoltageLine*. Die Berechnung der zur Bestimmung der Geräuschemissionen nötigen elektrischen Randfeldstärken der einzelnen Leiterseile geschieht in der Klasse *ElectricField*. Schlussendlich werden innerhalb der Klasse *AudibleNoise* anhand des elektrischen Feldes und der entsprechenden Regenraten die verschiedenen Schallpegel berechnet.

Abbildung 1.1 zeigt die Benutzeroberfläche der Applikation *HVLNoise*. Die Benutzerführung wird dabei in drei Schritte eingeteilt:

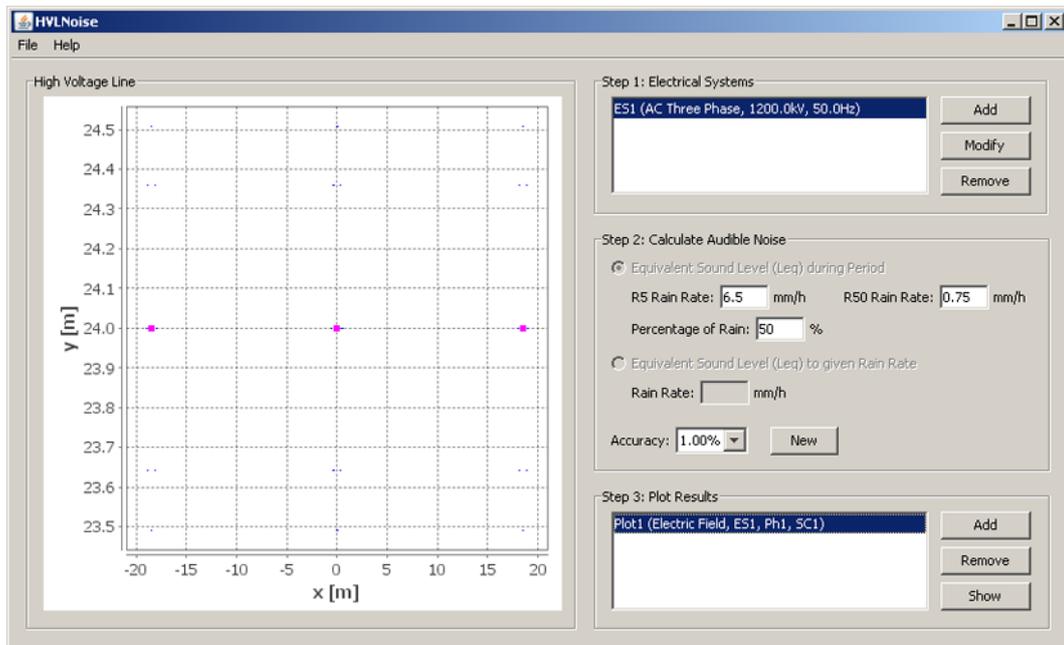


Abbildung 1.1: Benutzeroberfläche von *HVLNoise*

1. Zuerst wird der Benutzer aufgefordert die Geometrie der zu evaluierenden Hochspannungsfreileitung zu spezifizieren. Die bereits vorhandenen elektrischen Systeme werden ihm in einer Liste übersichtlich dargestellt. Eine nachträgliche Änderung oder die Entfernung eines bestehenden Systems ist jederzeit möglich. Die aktuell gewählte Leitungsgeometrie ist aus der Grafik auf der linken Seite ersichtlich.

¹<http://math.nist.gov/javanumerics/jama/>

²<http://www.jfree.org/jfreechart/>

1 Einleitung

2. Nun wählt der User den entsprechenden Berechnungsmodus für die akustischen Geräuschemissionen und startet dann die Berechnung der elektrischen Randfeldstärken sowie der Schallpegel.
3. Der letzte Schritt dient dem Benutzer zur Visualisierung der erzielten Rechenergebnisse. Zum einen können die Schallpegel auf einer bestimmten Höhe über Boden in horizontaler Richtung, zum anderen aber auch die elektrischen Randfeldstärken über den Umfangswinkel der einzelnen Leiterseile grafisch dargestellt werden. Eine Liste vereinfacht auch hier das Hinzufügen sowie Entfernen von Plots. Über die Menüleiste können zudem die Daten der zur Liste hinzugefügten Plots als Komma getrennte Werte (Format .csv) für eine weitere Verwendung exportiert werden.

2 Eingabe und Speicherung der Leitungsparameter

Im ersten Schritt erfolgt die Spezifizierung der Geometrie der Hochspannungsfreileitung durch den Benutzer. Die Applikation beschränkt sich auf drei mögliche elektrische Systeme: Dreiphasige Wechselstromsysteme mit einer Frequenz von 50 Hz wie im europäischen Energieversorgungsnetz, einphasige Wechselstromsysteme mit einer Frequenz von $16\frac{2}{3}$ Hz wie im Bahnstromnetz der Schweiz, von Deutschland oder Österreich sowie zweiphasige (bipolare) Gleichstromsysteme.

2.1 Benutzeroberfläche zur Eingabe der Leitungsparameter

Zur Eingabe der Leitungsparameter fügt der Benutzer im Hauptfenster von *HVLNoise* ein neues elektrisches System hinzu oder wählt ein bestehendes zur Modifikation aus. Im folgenden Dialogfenster, welches in Abbildung 2.1 dargestellt ist, können die Parameter des elektrischen Systems angegeben werden.

Zunächst wählt der Benutzer zwischen den drei möglichen elektrischen Systemen und spezifiziert die entsprechenden Systemspannungen in kV. Im Dreiphasenwechselstromsystem erfolgt die Angabe der Spannung als RMS-Wert der verketteten Spannung, im Einphasen-Bahnstromsystem als RMS-Wert der Phasenspannung. Im Fall von Gleichstromsystemen können die Phasenspannungen der zwei Leiter unabhängig voneinander angegeben werden. So ist es möglich, auch Geräuschemissionen bei Erdung eines Leiters zu untersuchen.

Für die Platzierung der Leiter resp. der dem Leiter zugehörigen Teilleiter werden xy-Koordinatensysteme eingeführt wie in Abbildung 2.2 gezeigt. Im Folgenden wird davon ausgegangen, dass die Geometrie in Längsrichtung der Hochspannungsleitung näherungsweise konstant bleibt, das heisst der Durchhang der Leiterseile wird vernachlässigt, was schliesslich auf eine reduzierte, zweidimensionale Beschreibung führt.

Im ersten Koordinatensystem werden die x- und y-Positionen der Leitermittelpunkte in m angegeben. Da sich der Boden bei $y = 0$ befindet, muss die y-Koordinate des Leiters einen positiven Wert aufweisen. In einem zweiten, untergeordneten Koordinatensystem (Ursprung ist der Leitermittelpunkt) kann die Anordnung der Teilleiter vorgenommen werden, welche für alle Leiter des entsprechenden Systems identisch ist. Dabei wird zwischen zwei Eingabemodi unterschieden: dem Standard- sowie dem Expertenmodus. Unabhängig vom gewählten Modus wird die aktuell spezifizierte Teilleitergeometrie dem Benutzer grafisch dargestellt wie aus Abbildung 2.1 ersichtlich ist.

2 Eingabe und Speicherung der Leitungsparameter

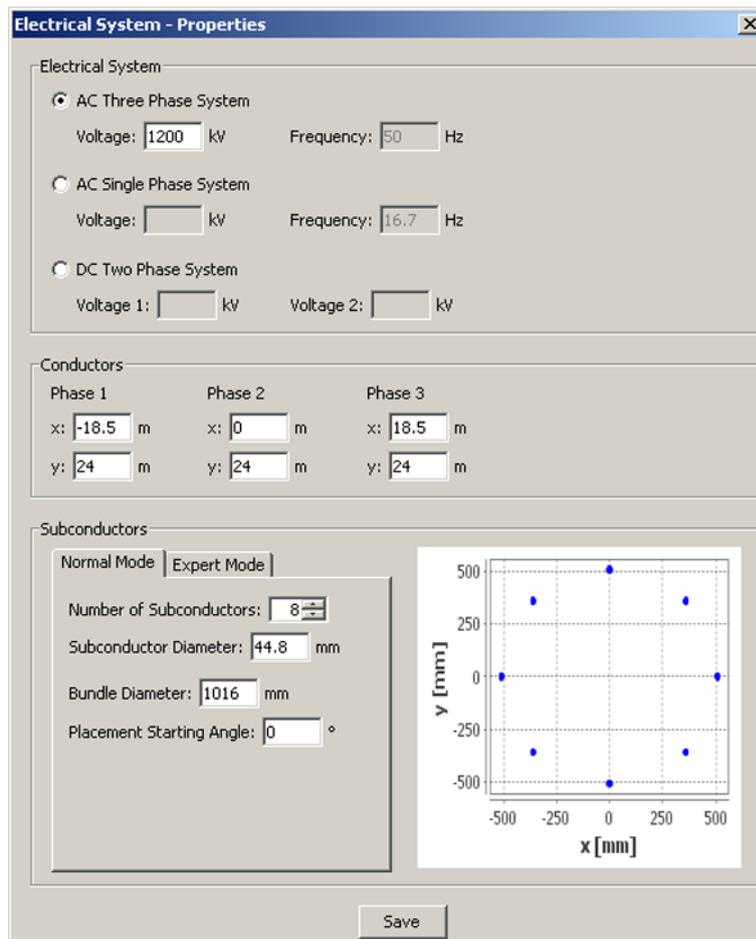


Abbildung 2.1: Dialogfenster zur Angabe der Parameter eines elektrischen Systems in der Anwendung

Im Standardmodus erfolgt eine äquidistante Platzierung von n Teilleitern gleichen Durchmessers d (in mm) auf einem Kreis mit Bündeldurchmesser D (in mm) entgegen dem Uhrzeigersinn. Über den Startwinkel α (in $^\circ$) kann die Position des ersten Teilleiters bestimmt werden. Die Berechnungen der zum Leitermittelpunkt relativen x - und y -Positionen des i -ten Teilleiters werden durch die Gleichungen (2.1) und (2.2) beschrieben.

$$x_i = \frac{D}{2} \cos \left(\frac{2\pi}{n}(i-1) + \alpha \frac{\pi}{180^\circ} \right) \quad i = 1, 2, \dots, n \quad (2.1)$$

$$y_i = \frac{D}{2} \sin \left(\frac{2\pi}{n}(i-1) + \alpha \frac{\pi}{180^\circ} \right) \quad i = 1, 2, \dots, n \quad (2.2)$$

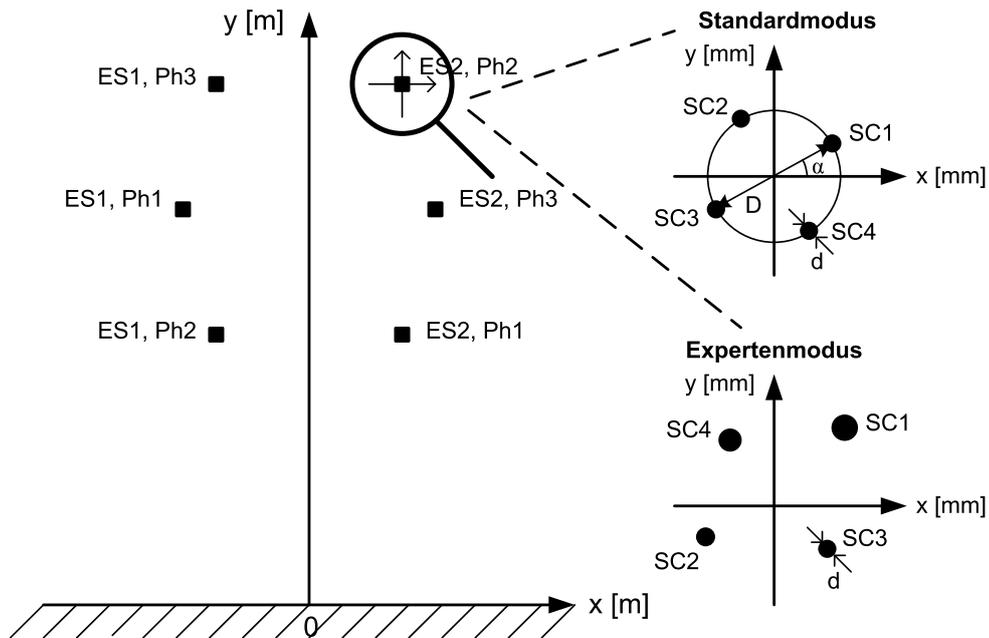


Abbildung 2.2: xy-Koordinatensysteme zur Platzierung der Leiter resp. Teilleiter

Der Expertenmodus erlaubt es dem Benutzer sämtliche Teilleiter unabhängig voneinander relativ zum Leitermittelpunkt unter Angabe von x- und y-Position (in mm) und unterschiedlichen Leiterdurchmessern d (in mm) zu platzieren. Die gespeicherten Teilleiter sind in einer Liste ersichtlich und können jederzeit modifiziert oder entfernt werden. Dies zeigt Abbildung 2.3.

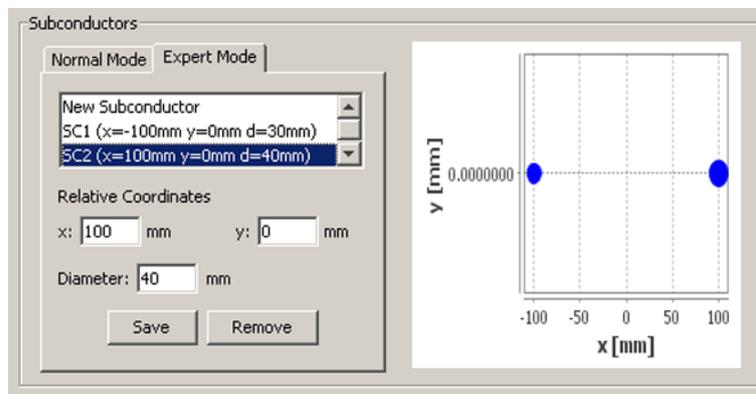


Abbildung 2.3: Ansicht des Expertenmodus zur Platzierung der Teilleiter

2.2 Hierarchische Datenstruktur zur Speicherung der Leitungsparameter

Um die Leitungsparameter nach der Eingabe durch den Benutzer für die weitere Verwendung übersichtlich und strukturiert zu speichern, werden vier Java-Klassen eingeführt. Die Hochspannungsfreileitung mit all ihren elektrischen Systemen wird als Instanz der Klasse *HighVoltage-*

2 Eingabe und Speicherung der Leitungsparameter

Line gespeichert, welche wiederum Instanzen der Klasse *ElectricalSystem* hält. Die Instanzen von *ElectricalSystem* definieren ihrerseits wiederum Instanzen der Klasse *Conductor* und diese wiederum Instanzen der Klasse *Subconductor*. Somit entsteht eine hierarchische Datenstruktur wie sie in Abbildung 2.4 gezeigt ist.

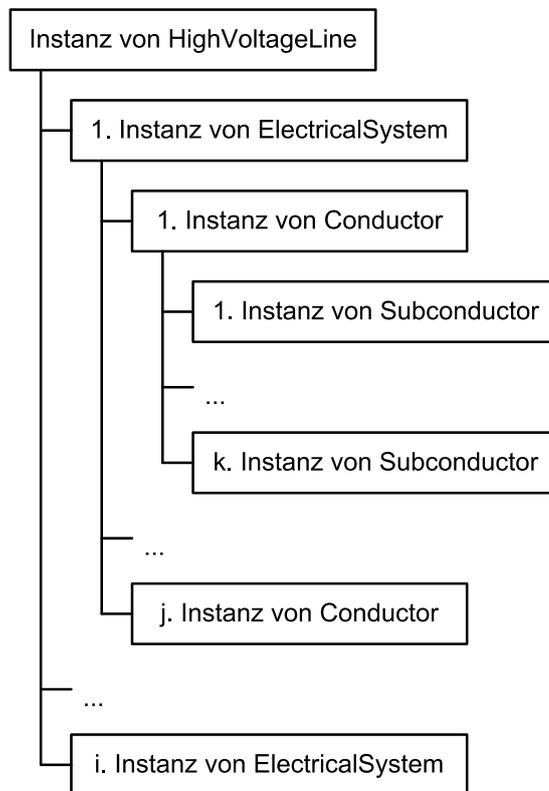


Abbildung 2.4: Hierarchische Datenstruktur zur Speicherung der Leitungsparameter

2.2.1 Hochspannungsfreileitung (Klasse *HighVoltageLine*)

Die Klasse *HighVoltageLine* soll im Wesentlichen die im Hauptfenster der Anwendung (siehe Abbildung 1.1) als Liste ersichtlichen elektrischen Systeme verwalten. Die dazu nötigen Operationen sind das Hinzufügen, das Modifizieren und das Entfernen von elektrischen Systemen. Bei der Ausführung einer solchen Operation soll zudem die grafische Darstellung der Hochspannungsfreileitung entsprechend angepasst werden.

Variablendeklarationen

Innerhalb der Klasse werden Variablen für die Speicherung der elektrischen Systeme sowie der Anzahl der Dreiphasenwechselstrom-, Einphasenwechselstrom- und Gleichstromsysteme deklariert. Die Instanzen von *ElectricalSystem* werden dabei in einer *ArrayList* abgelegt. Hierbei handelt es sich um ein dynamisches Array, dessen Speicherbedarf zu Beginn nicht bekannt sein muss. Würden konventionelle Arrays benutzt, müsste die vom Benutzer gewünschte Anzahl Systeme speziell abgefragt werden, um daraufhin ein Array entsprechender Grösse erstellen

zu können. Weiter werden für die grafische Darstellung der Hochspannungsfreileitung einige Variablen deklariert, auf die aber nicht weiter eingegangen werden soll.

Konstruktor

Der während der Instanziierung eines Objekts der Klasse ausgeführte Konstruktor beinhaltet lediglich Variablendefinitionen. Die bereits erwähnte *ArrayList* zur Speicherung der elektrischen Systeme wird instanziiert sowie die Zähler für die Anzahl Systeme der drei verschiedenen Typen zu Null gesetzt. Zudem werden weitere grafikspezifische Variablen definiert.

Hinzufügen und Modifizieren von elektrischen Systemen

Für das Hinzufügen und Modifizieren von elektrischen Systemen werden drei Methoden eingeführt, das heisst eine pro Systemtyp. Die Argumente der Funktionen unterscheiden sich lediglich bei der Angabe der Spannung. Während für Dreiphasen- sowie Einphasenwechselstromsysteme ein Parameter genügt, müssen für Gleichstromsysteme zwei Spannungsangaben möglich sein. Die Methoden *addOrSetAcThreePhaseSystem(..)*, *addOrSetAcSinglePhaseSystem(..)* und *addOrSetDcTwoPhaseSystem(..)* übernehmen die nachfolgend aufgelisteten Argumente:

- **double acSystemVoltage** in kV
RMS-Wert der Aussenleiter- resp. Phasenspannung bei Dreiphasen- resp. Einphasenwechselstromsystemen.
- **double[] dcSystemVoltage** in kV
Array mit zwei Phasenspannungen des Gleichstromsystems.
- **Matrix conductorPositions** in m
Matrix mit zwei Zeilen für x- und y-Positionen der Leitermittelpunkte. Für Drei-, Ein- und Zweiphasensysteme sind die Dimensionen (2×3) , (2×1) und (2×2) .
- **Matrix relSubconductorPositions** in mm
Matrix mit zwei Zeilen für relative x- und y-Positionen der Teilleiter.
- **Matrix subconductorRadii** in mm
Zeilenvektor mit den Radien der einzelnen Teilleiter.
- **boolean subconductorNormalModeEnabled**
Gibt an, ob der Standardmodus zur Eingabe der Teilleiter aktiviert ist.
- **double[] subconductorNormalModeParameters** in mm und $^\circ$
Bündeldurchmesser und Startwinkel zur Platzierung der Teilleiter im Standardmodus.
- **int systemIndex**
–1 für das Hinzufügen eines Systems, ansonsten ist der Index des Systems innerhalb der *ArrayList* zu verstehen, welches ersetzt werden soll.

2 Eingabe und Speicherung der Leitungsparameter

Innerhalb der Methoden wird jeweils ein neues Objekt der Klasse *ElectricalSystem* erzeugt, das entsprechende System durch Aufruf der *create*-Funktion (siehe Abschnitt 2.2.2) angelegt und abhängig vom Übergabewert von *systemIndex* an die *ArrayList* als neues Element angefügt oder als Ersatz für ein bestehendes in die *ArrayList* aufgenommen. Zusätzlich wird beim Hinzufügen eines neuen Systems der jeweilige Zähler des Systemtyps inkrementiert. Schlussendlich werden Funktionen zur Aktualisierung der grafischen Darstellung der Hochspannungsfreileitung aufgerufen.

Entfernen von elektrischen Systemen

Durch Übergabe der Position eines elektrischen Systems innerhalb der *ArrayList* kann über die Funktion *removeSystem(..)* ein System entfernt werden. Dabei wird der dem Systemtyp zugehörige Zähler dekrementiert. Anschliessend werden die Geometriedaten der Hochspannungsfreileitung auf der grafischen Oberfläche neu gezeichnet.

Weitere Funktionalitäten

Weitere in der Klasse *HighVoltageLine* zur Verfügung stehende Funktionen (ohne Methoden zur grafischen Darstellung) sind im Folgenden aufgeführt:

- **ElectricalSystem getSystem(int index)**
Gibt das elektrische System an entsprechender Stelle in der *ArrayList* zurück.
- **int getSystemIndex(ElectricalSystem system)**
Die Position des übergebenen Systems in der *ArrayList* wird ermittelt.
- **ElectricalSystem[] getAcThreePhaseSystems()**
Gibt alle Dreiphasenwechselstromsysteme als Array zurück.
- **int[] getAcThreePhaseSystemIndices()**
Die Indizes aller Dreiphasenwechselstromsysteme bezogen auf die *ArrayList* werden zurückgegeben.
- **ElectricalSystem[] getAcSinglePhaseSystems()**
Gibt alle Einphasenwechselstromsysteme als Array zurück.
- **int[] getAcSinglePhaseSystemIndices()**
Die Indizes aller Einphasenwechselstromsysteme bezogen auf die *ArrayList* werden zurückgegeben.
- **ElectricalSystem[] getDcTwoPhaseSystems()**
Gibt alle Gleichstromsysteme als Array zurück.
- **int[] getDcTwoPhaseSystemIndices()**
Die Indizes aller Gleichstromsysteme bezogen auf die *ArrayList* werden zurückgegeben.
- **int getNumberOfSystems()**
Gibt die Anzahl vorhandener elektrischer Systeme zurück.

2.2.2 Elektrische Systeme (Klasse *ElectricalSystem*)

Die Klasse *ElectricalSystem* dient der Definition eines elektrischen Systems. Mittels dieser Klasse sollen die drei möglichen Systemtypen durch ihre Parameter beschrieben werden können. Die zur Verfügung stehenden Funktionen beschränken sich im Wesentlichen auf die Erzeugung eines Systems sowie das Abfragen von Systemparametern. Der während der Instanziierung eines Objekts von *ElectricalSystem* ausgeführte Konstruktor enthält keinen spezifischen Quellcode.

Variablendeklarationen

Für die Beschreibung des elektrischen Systems werden grundsätzlich drei Variablen benötigt. Zunächst werden Variablen für Systemtyp sowie die Betriebsspannung deklariert, wobei zwischen zwei Spannungsvariablen je nach Wechsel- oder Gleichstromsystem unterschieden wird. Die Geometrie des Systems wird schliesslich in einem *Array*, welches Instanzen der Klasse *Conductor* hält, gespeichert. Weitere Variablen bestimmen den gewählten Eingabemodus der Teilleiter (Standard- oder Expertenmodus) sowie bei aktiviertem Standardmodus den Bündeldurchmesser sowie den Startwinkel zur Platzierung der Teilleiter.

Erstellen eines elektrischen Systems

Das Setzen der Systemparameter resp. das Konfigurieren der Instanz von *ElectricalSystem* erfolgt nicht direkt über den Konstruktor. Vielmehr werden die drei Funktionen *createAcThreePhaseSystem(..)*, *createAcSinglePhaseSystem(..)* sowie *createDcTwoPhaseSystem(..)* zur Erstellung von Dreiphasenwechselstrom-, Einphasenwechselstrom- und Zweiphasengleichstromsystemen eingeführt. Die Argumente sind genau die selben wie die der *addOrSet*-Funktionen der Klasse *HighVoltageLine* (siehe Abschnitt 2.2.1).

Innerhalb der *create*-Funktionen werden die zuvor deklarierten Variablen für den Systemtyp sowie die Systemspannung entsprechend der übergebenen Parameter initialisiert. In einem nächsten Schritt werden je nach Systemtyp die Anzahl erforderlichen *Conductor*-Instanzen erstellt. Als Argumente des Konstruktors werden die Position des Leitermittelpunktes, die relativen Teilleiterpositionen sowie die Teilleiterdurchmesser übergeben. Zuletzt werden die Objekte im zuvor deklarierten *Array* gespeichert. An dieser Stelle soll angemerkt werden, dass es sich hierbei um ein *Array* fixer Grösse handelt, da die Anzahl Leiter bei der Erzeugung des Systems bekannt ist.

Zum Schluss werden die Variablen zur Speicherung des Teilleiter-Eingabemodus sowie des Bündeldurchmessers und des Startwinkels zur Platzierung initialisiert.

Abfragen der Systemparameter

Zur Abfrage von Parametern des elektrischen Systems werden diverse Funktionen eingeführt. Diese können in zwei Gruppen unterteilt werden. Zum einen sind dies Methoden, welche systemspezifische Parameter zurückgeben, und zum anderen Funktionen, die Geometriedaten des Systems auslesen. Im Folgenden sind die Methoden zur Abfrage von systembezogenen Parametern angegeben:

2 Eingabe und Speicherung der Leitungsparameter

- **SystemType getSystemType()**
Gibt den Typ des elektrischen Systems als Instanz von *SystemType* zurück.
- **double getACSystemVoltage()**
Liefert bei Wechselstromsystemen die Betriebsspannung zurück. Im Fall von Dreiphasensystemen ist dies der RMS-Wert der verketteten Spannung, bei Einphasensystemen der RMS-Wert der Phasenspannung.
- **getDCSystemVoltage(int conductorIndex)**
Unter Angabe des Leiterindex (0 oder 1) gibt die Funktion die Gleichspannung der entsprechenden Phase bei Gleichstromsystemen zurück.
- **int getNumberOfConductors()**
Gibt die Anzahl Leiter des elektrischen Systems zurück. Die möglichen Rückgabewerte (1, 2, 3) werden durch die Systemtypen definiert.

Die nachfolgend aufgelisteten Funktionen dienen der Abfrage von Geometriedaten des elektrischen Systems:

- **int getNumberOfSubconductors()**
Gibt die Anzahl Teilleiter pro Phase zurück. Wie bereits erwähnt bleibt die Leitergeometrie innerhalb eines Systems konstant.
- **double getConductorAbsolutePositionX(int conductorIndex)**
Liefert die absolute x-Position (in m) des durch den übergebenen Index gekennzeichneten Leiters.
- **double getConductorAbsolutePositionY(int conductorIndex)**
Liefert die absolute y-Position (in m) des durch den übergebenen Index gekennzeichneten Leiters.
- **Matrix getConductorAbsolutePosition(int conductorIndex)**
Im Gegensatz zu den zwei obigen Methoden gibt diese die absolute x- und y-Position (in m) des entsprechenden Leiters in Form des Ortsvektors (Spaltenvektor) zurück.
- **double getSubconductorAbsolutePositionX(int conductorIndex, int subconductorIndex)**
Gibt die absolute x-Position (in m) des durch die Indizes spezifizierten Teilleiters zurück.
- **double getSubconductorAbsolutePositionY(int conductorIndex, int subconductorIndex)**
Gibt die absolute y-Position (in m) des durch die Indizes spezifizierten Teilleiters zurück.
- **Matrix getSubconductorAbsolutePosition(int conductorIndex, int subconductorIndex)**
Im Gegensatz zu den zwei obigen Methoden gibt diese die absolute x- und y-Position (in m) des entsprechenden Teilleiters in Form des Ortsvektors (Spaltenvektor) zurück.
- **double getSubconductorRelativePositionX(int subconductorIndex)**
Liefert die x-Position (in mm) relativ zum Leitermittelpunkt eines Teilleiters. Der Index des Teilleiters bezieht sich auf die Position innerhalb eines Leiters.

- **double getSubconductorRelativePositionY(int subconductorIndex)**
Liefert die y-Position (in mm) relativ zum Leitermittelpunkt eines Teilleiters. Der Index des Teilleiters bezieht sich auf die Position innerhalb eines Leiters.
- **double getSubconductorRadius(int subconductorIndex)**
Gibt den Radius (in mm) des Teilleiters an entsprechender Position innerhalb eines Leiters zurück.

Weitere Funktionalitäten

Weitere in der Klasse *ElectricalSystem* zur Verfügung stehende Funktionen sind im Folgenden aufgeführt:

- **boolean isSubconductorNormalModeEnabled()**
Liefert *true*, wenn der Standardmodus zur Eingabe der Teilleiter für das System aktiviert ist, ansonsten *false*.
- **double getSubconductorNormalModeParameter(int parameterIndex)**
Im Teilleiter-Standardmodus kann über den Parameterindex für 0 der Bündeldurchmesser (in mm) und für 1 der Startwinkel zur Platzierung der Teilleiter (in °) zurückgegeben werden.

2.2.3 Leiter (Klasse *Conductor*)

Zur Speicherung der Leitergeometrie wird die Klasse *Conductor* eingeführt. Ihre Hauptfunktionalitäten liegen in der Definition des Leitermittelpunktes sowie der vorhandenen Teilleiter. Die Parameter zur Erzeugung einer *Conductor*-Instanz werden direkt dem Konstruktor übergeben.

Variablendeklarationen

Gemäss den zwei Hauptfunktionen der Klasse werden Variablen zur Speicherung des Leitermittelpunktes sowie der Teilleiter benötigt. Die Positionsvariable des Leiters wird als Ortsvektor (Spaltenvektor) mit der x-Position auf der ersten Zeile und der y-Position auf der zweiten Zeile deklariert. Die Parameter der einzelnen Teilleiter innerhalb des Leiters werden schliesslich in einem *Array* von Instanzen der Klasse *Subconductor* gehalten.

Erstellen eines Leiters (Konstruktor)

Das Erzeugen einer *Conductor*-Instanz geschieht direkt durch den Aufruf des Konstruktors mit den entsprechenden Argumenten. Für die Konfiguration des Objekts ist kein weiterer Funktionsaufruf, wie dies bei den Klassen *HighVoltageLine* und *ElectricalSystem* der Fall war, nötig. Bei der Erstellung eines Leiterobjekts sind dem Konstruktor die Parameter gemäss folgender Auflistung zu übergeben:

2 Eingabe und Speicherung der Leitungsparameter

- **Matrix positionVector** in m
Ortsvektor des Leitermittelpunktes mit x-Koordinate in der ersten und y-Koordinate in der zweiten Zeile des Spaltenvektors.
- **Matrix relSubconductorPositions** in mm
Matrix mit zwei Zeilen für relative x- und y-Positionen der Teilleiter (wie in *HighVoltageLine* und *ElectricalSystem*).
- **Matrix subconductorRadii** in mm
Zeilenvektor mit den Radien der einzelnen Teilleiter (wie in *HighVoltageLine* und *ElectricalSystem*).

Innerhalb des Konstruktors wird die Positionsvariable des Leiters mittels übergebenem Ortsvektor definiert. Danach werden die Instanzen der Klasse *Subconductor* erzeugt. Bei der Instanziierung wird der Leitermittelpunkt, die relative Teilleiterposition sowie der Teilleiterradius benötigt. Die Teilleiterobjekte werden schlussendlich in dem dafür vorgesehenen *Array* gespeichert.

Abfragen der Leiterparameter

Auch im Fall der Klasse *Conductor* sind verschiedenste Funktionen zum Abruf der Leiterparameter erforderlich. Viele dieser Methoden werden von Methoden der übergeordneten Klasse *ElectricalSystem* angesprochen. Nachfolgend sind die Funktionen einzeln beschrieben:

- **double getAbsolutePositionX()**
Gibt die absolute x-Position (in m) des Leitermittelpunktes zurück.
- **double getAbsolutePositionY()**
Gibt die absolute y-Position (in m) des Leitermittelpunktes zurück.
- **Matrix getAbsolutePosition()**
Im Gegensatz zu den zwei obigen Methoden gibt diese die absolute x- und y-Position (in m) des entsprechenden Leitermittelpunktes in Form des Ortsvektors (Spaltenvektor) zurück.
- **int getNumberOfSubconductors()**
Liefert die Anzahl Teilleiter, die im Leiter vorhanden sind.
- **double getSubconductorAbsolutePositionX(int subconductorIndex)**
Gibt die absolute x-Position (in m) des durch den Index spezifizierten Teilleiters zurück.
- **double getSubconductorAbsolutePositionY(int subconductorIndex)**
Gibt die absolute y-Position (in m) des durch den Index spezifizierten Teilleiters zurück.
- **Matrix getSubconductorAbsolutePosition(int subconductorIndex)**
Im Gegensatz zu den zwei obigen Methoden gibt diese die absolute x- und y-Position (in m) des entsprechenden Teilleiters in Form des Ortsvektors (Spaltenvektor) zurück.
- **double getSubconductorRelativePositionX(int subconductorIndex)**
Liefert die x-Position (in mm) relativ zum Leitermittelpunkt des durch den Index spezifizierten Teilleiters.

- **double getSubconductorRelativePositionY(int subconductorIndex)**
Liefert die y-Position (in mm) relativ zum Leitermittelpunkt des durch den Index spezifizierten Teilleiters.
- **double getSubconductorRadius(int subconductorIndex)**
Gibt den Radius (in mm) des Teilleiters an entsprechender Position innerhalb des Leiters zurück.

2.2.4 Teilleiter (Klasse *Subconductor*)

Die letzte Klasse, die zur Speicherung der Leitungsparameter notwendig ist, wird *Subconductor* genannt. Instanzen dieser Klasse beschreiben die Beschaffenheit der einzelnen Leiterseile der Hochspannungsfreileitung, das heisst die absolute und relative Position sowie den Radius. Sie sind hierarchisch auf der untersten Ebene angesiedelt (siehe Abbildung 2.4).

Variablendeklarationen

Zunächst werden Variablen zur Speicherung der absoluten (in m) und relativen (in mm) Position sowie des Radius (in mm) des Teilleiters angelegt. Wie bereits bei der Deklaration der Leitermittelpunktvariablen in *Conductor* sollen auch die Teilleiterpositionen als Ortsvektoren (Spaltenvektoren) mit der x-Position auf der ersten Zeile und der y-Position auf der zweiten Zeile implementiert werden.

Erstellen eines Teilleiters (Konstruktor)

Das Anlegen eines Teilleiters erfolgt bei der Erstellung einer Instanz der Klasse *Subconductor* durch Übergabe der Argumente an den Konstruktor. Diese sind die absolute und relative Teilleiterposition sowie der Teilleiterradius:

- **Matrix conductorPosition** in m
Ortsvektor des Leitermittelpunktes des übergeordneten Leiters mit x-Koordinate in der ersten und y-Koordinate in der zweiten Zeile des Spaltenvektors.
- **Matrix relPositionVector** in mm
Spaltenvektor mit zwei Zeilen für relative x- und y-Position des Teilleiters bezogen auf den Leitermittelpunkt des übergeordneten Leiters.
- **double radius** in mm
Radius des Teilleiters.

Beim Aufruf des Konstruktors werden die zuvor deklarierten Variablen initialisiert. Die Zuweisung der relativen Teilleiterposition sowie des Teilleiterradius erfolgt direkt durch die entsprechend übergebenen Argumente. Zudem wird die absolute Teilleiterposition durch Vektoraddition des Ortsvektors des Leitermittelpunktes mit dem relativen Ortsvektor des Teilleiters gebildet.

Abfragen der Teilleiterparameter

Wie die Klasse *Conductor* stellt auch die Klasse *Subconductor* elementare Funktionen zum Abruf der Teilleiterparameter bereit. Diese werden hauptsächlich durch Methoden der übergeordneten Klassen aufgerufen. Im Folgenden sind die vorhandenen Funktionen aufgelistet:

- **double getAbsolutePositionX()**
Gibt die absolute x-Position (in m) des Teilleiters zurück.
- **double getAbsolutePositionY()**
Gibt die absolute y-Position (in m) des Teilleiters zurück.
- **Matrix getAbsolutePosition()**
Im Gegensatz zu den zwei obigen Methoden gibt diese die absolute x- und y-Position (in m) des Teilleiters in Form des Ortsvektors (Spaltenvektor) zurück.
- **double getRelativePositionX()**
Liefert die relative x-Position (in mm) des Teilleiters.
- **double getRelativePositionY()**
Liefert die relative y-Position (in mm) des Teilleiters.
- **double getRadius()**
Gibt den Radius (in mm) des Teilleiters zurück.

2.3 Schlussfolgerungen

Mit der grafischen Oberfläche zur Eingabe der Leitungsparameter und der hierarchischen Datenstruktur zur Speicherung der Hochspannungsfreileitung ist nun die Basis für eine weiterführende Datenverarbeitung gelegt.

Dem Benutzer steht eine komfortable Oberfläche zur Spezifizierung der elektrischen Systeme zur Verfügung. Je nach Erfordernis der Leiterkonfiguration kann zwischen dem Standard- resp. Expertenmodus zur Platzierung der Teilleiter gewechselt werden. Zudem bietet die Benutzeroberfläche von *HVLNoise* eine Visualisierung der Leitungsgeometrie an, welche bei Anpassungen der Parameter aktualisiert und somit dem User immer die aktuell eingegebene Hochspannungsfreileitung dargestellt wird.

Über eine einzige Instanz der Klasse *HighVoltageLine* werden sämtliche elektrische Systeme mit all ihren Parametern durch die Benutzeroberfläche von *HVLNoise* verwaltet. Die in der Hierarchie untergeordneten Klassen wie *ElectricalSystem*, *Conductor* sowie *Subconductor* werden dabei nur durch übergeordnete Klassen instanziiert. Zugriffe zur Modifikation oder Abfrage von Parametern der Hochspannungsfreileitung geschehen also ausschliesslich über das *HighVoltageLine*-Objekt. Für die weiterführende Datenverarbeitung sind die Methoden von *HighVoltageLine* entscheidend, denn sie bilden die zentrale Schnittstelle zu Parametern der Hochspannungsfreileitung. Gewisse Funktionen geben Instanzen von *ElectricalSystem* zurück, sodass ein Teil der Zugriffe nur indirekt über *HighVoltageLine* erfolgt. Eine weitere wichtige Schnittstelle zur Abfrage von Systemdaten bilden also auch die Methoden der Klasse *ElectricalSystem*.

3 Berechnung der elektrischen Randfeldstärken mittels Ersatzladungsverfahren

Die akustischen Geräuschemissionen von Hochspannungsfreileitungen hängen zu einem wesentlichen Teil vom Verlauf des elektrischen Feldes entlang des Umfangs der Leiterseile ab. Als Voraussetzung zur Vorhersage dieser Geräuschemissionen müssen also die elektrischen Randfeldstärken (RMS-Werte) sämtlicher Teilleiter bekannt sein. Zur Feldberechnung bietet sich aus Gründen der einfachen Elektrodengeometrien (jeder Teilleiter bildet einen unendlich langen Zylinder) das Ersatzladungsverfahren an. Dieses erfordert zudem gegenüber einer Berechnung mit finiten Elementen keine Diskretisierung des Raumes.

3.1 Mathematische Formulierung des Ersatzladungsverfahrens

Wie bereits bei der Eingabe der Leitungsparameter erwähnt (siehe Abschnitt 2.1) erfolgt die Beschreibung der Hochspannungsfreileitung lediglich in der Querschnittsebene. Zur Berechnung der elektrischen Feldstärken werden die Leiterseile als unendlich lange Zylinder betrachtet auf deren Umfang die Randbedingungen in Form des jeweils auftretenden (zeitabhängigen) elektrischen Potentials zu erfüllen sind.

Zur Anwendung des Ersatzladungsverfahrens werden entlang der Teilleiterumrandung Konturpunkte in äquidistantem Abstand platziert, deren Ladungen $q_{K,i}$ sowie deren elektrische Potentiale $\phi_{K,i}$ gegeben sind. Im Innern des Leiterseils wird zu jedem Konturpunkt eine Linienladung unendlicher Länge, die sogenannte Ersatzladung $q_{L,i}$ definiert, deren Grösse sich durch die Randbedingungen bestimmen lässt. Abbildung 3.1 zeigt die beschriebene Anordnung der Konturpunkte sowie Ersatzladungen im xy-Koordinatensystem.

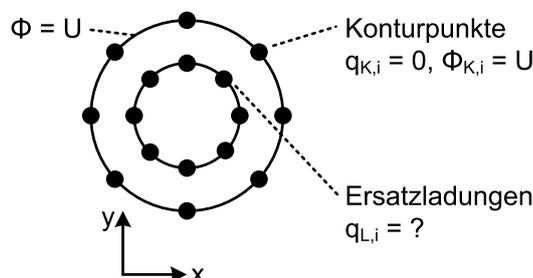


Abbildung 3.1: Anordnung von Konturpunkten und Ersatzladungen auf Teilleiterkontur

3 Berechnung der elektrischen Randfeldstärken mittels Ersatzladungsverfahren

Für die mathematische Formulierung des Ersatzladungsverfahrens mit n Konturpunkten bzw. Ersatzladungen seien zwei Potentialvektoren gemäss (3.1) definiert. Dabei bezeichnet i den Laufindex über die Konturpunkte resp. Ersatzladungen.

$$[\phi_{K,i}] = \begin{pmatrix} \phi_{K,1} \\ \phi_{K,2} \\ \vdots \\ \phi_{K,n} \end{pmatrix} \quad [\phi_{L,i}] = \begin{pmatrix} \phi_{L,1} \\ \phi_{L,2} \\ \vdots \\ \phi_{L,n} \end{pmatrix} \quad (3.1)$$

Der Index K steht für das elektrische Potential des Konturpunktes, der Index L für das Potential der Ersatzladung. Analog dazu werden Ladungsvektoren nach (3.2) eingeführt. Der Index K beschreibt die Ladung des Konturpunktes, der Index L die Ersatzladung.

$$[q_{K,i}] = \begin{pmatrix} q_{K,1} \\ q_{K,2} \\ \vdots \\ q_{K,n} \end{pmatrix} \quad [q_{L,i}] = \begin{pmatrix} q_{L,1} \\ q_{L,2} \\ \vdots \\ q_{L,n} \end{pmatrix} \quad (3.2)$$

Unter Berücksichtigung der obigen Definitionen lässt sich nun das Ersatzladungsverfahren mittels Matrixgleichung

$$\begin{pmatrix} [\phi_{K,i}] \\ [\phi_{L,i}] \end{pmatrix} = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix} \begin{pmatrix} [q_{K,i}] \\ [q_{L,i}] \end{pmatrix} \quad (3.3)$$

beschreiben. Jedes Potential $\phi_{K,i}$ oder $\phi_{L,i}$ entsteht durch die Superposition der elektrischen Potentiale aller Ersatzladungen $q_{K,i}$ sowie $q_{L,i}$.

Gegeben ist dabei das elektrische Potential $[\phi_{K,i}]$ in den Konturpunkten gemäss dem Verlauf der entsprechenden Phasenspannung des Teileiters. Die Ladungen der Konturpunkte werden zu Null gesetzt, das heisst $[q_{K,i}] = [0]$. Damit erfolgt die Bestimmung der Ersatzladungen nach Gleichung (3.4) resp. (3.5).

$$[\phi_{K,i}] = P_{12} [q_{L,i}] \quad (3.4)$$

$$[q_{L,i}] = P_{12}^{-1} [\phi_{K,i}] \quad (3.5)$$

Die gesuchten Ersatzladungen $[q_{L,i}]$ besitzen die Einheit C/m, die elektrischen Potentiale $[\phi_{K,i}]$ die Einheit V. Die $(n \times n)$ -Matrix P_{12} wird Potentialkoeffizientenmatrix, ihre Inverse P_{12}^{-1} Kapazitätsmatrix genannt. Für die Einträge der Matrix P_{12} gilt $p_{ij} = p_{ji}$. Die Herleitung dieser Potentialkoeffizienten erfolgt in Abschnitt 3.3. Das elektrische Feld bildet sich schlussendlich durch Superposition der von den Ersatzladungen stammenden Teilfelder.

3.2 Platzierung der Konturpunkte und Ersatzladungen

Bei der Berechnung des elektrischen Feldes existiert neben den elektrischen Potentialen auf den Teileiterkonturen auch das Nullpotential des Erdbodens als Randbedingung. Um diese Forderung zu erfüllen, wird zu jeder Ersatzladung (Linienladung) eine an der x-Achse (Boden) gespiegelte Ladung entgegengesetzten Vorzeichens eingeführt, eine sogenannte Spiegelladung. Somit wird $\phi = 0$ auf dem Erdboden erreicht. Grundsätzlich geschieht die Platzierung der Konturpunkte und Ersatzladungen für einen Teileiter wie in Abbildung 3.2 gezeigt.

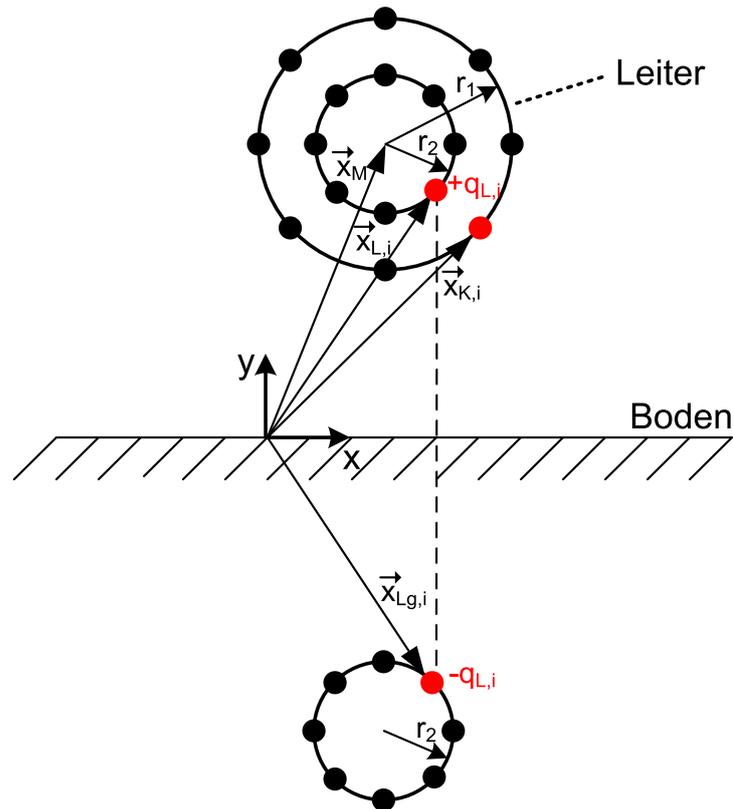


Abbildung 3.2: Platzierung von Konturpunkten und Ersatzladungen sowie Zuordnung der Spiegelladungen eines Teilleiters.

Die Konturpunkte werden äquidistant auf einem Kreis mit Radius des halben Teileiterdurchmessers angeordnet. Die gleiche Anzahl Ersatzladungen wird so innerhalb des Teilleiters positioniert, dass die Distanz zwischen Ladung und Konturpunkt etwa dem Abstand der Ladung zu benachbarten Ersatzladungen entspricht [3]. Aus dieser Forderung lässt sich über die Beziehung

$$r_1 - r_2 = 2 \cdot r_2 \sin\left(\frac{2\pi}{n} \cdot \frac{1}{2}\right) \quad (3.6)$$

3 Berechnung der elektrischen Randfeldstärken mittels Ersatzladungsverfahren

der Platzierungsradius r_2 für die Ersatzladungen gemäss

$$r_2 = \frac{r_1}{1 + 2 \sin\left(\frac{\pi}{n}\right)} \quad (3.7)$$

bestimmen, wobei r_1 den Teilleiterradius und n die Anzahl Konturpunkte auf der Teilleiterkontur bezeichnet.

Für die Platzierung der Konturpunkte sowie der Ersatzladungen gilt also in Abhängigkeit des Ortsvektors des Teilleitermittelpunktes \vec{x}_M und der Anzahl Teilleiter n folgende Gleichung:

$$\vec{x}_{K,i} = \vec{x}_{L,i} = \vec{x}_M + r \begin{pmatrix} \cos\left(\frac{2\pi}{n}(i-1)\right) \\ \sin\left(\frac{2\pi}{n}(i-1)\right) \end{pmatrix} \quad i = 1, 2, \dots, n \quad (3.8)$$

$\vec{x}_{K,i}$ und $\vec{x}_{L,i}$ bezeichnen die Ortsvektoren (Spaltenvektoren) der Konturpunkte resp. Ersatzladungen. Der Radius r ist entweder durch r_1 bei Positionierung der Konturpunkte oder durch r_2 bei Positionierung der Ersatzladungen zu ersetzen.

3.3 Herleitung der Potentialkoeffizienten

Im Folgenden sollen die zur Berechnung der Ersatzladungen benötigten Potentialkoeffizienten hergeleitet werden. Dazu wird eine Ersatzladung $+q_{L,j}$ und ihre Spiegelladung $-q_{L,j}$ entsprechend Abbildung 3.3 positioniert. Gesucht ist nun das elektrische Potential in jedem beliebigen Aufpunkt \vec{x} , welches durch die beiden Ladungen gebildet wird.

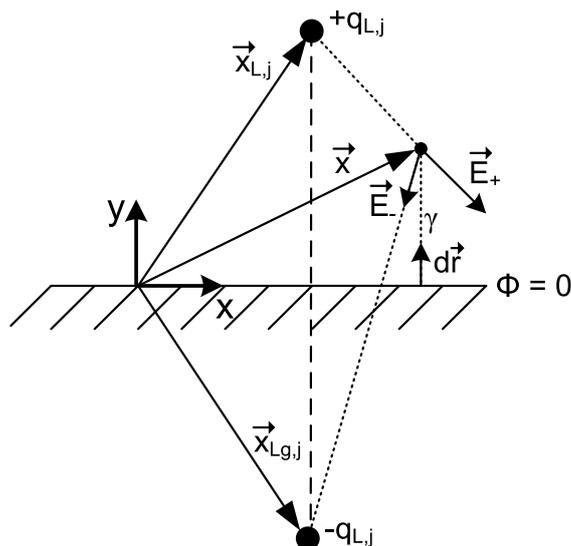


Abbildung 3.3: Ersatzladung und ihre Spiegelladung zur Herleitung der Potentialkoeffizienten

Die durch die beiden Linienladungen erzeugten elektrischen Felder lassen sich durch

$$\vec{E}_+ = \frac{q_{L,j}}{2\pi\epsilon_0\epsilon_r} \cdot \frac{\vec{x} - \vec{x}_{L,j}}{|\vec{x} - \vec{x}_{L,j}|^2} \quad (3.9)$$

und

$$\vec{E}_- = -\frac{q_{L,j}}{2\pi\epsilon_0\epsilon_r} \cdot \frac{\vec{x} - \vec{x}_{Lg,j}}{|\vec{x} - \vec{x}_{Lg,j}|^2} \quad (3.10)$$

beschreiben.

Das resultierende elektrische Feld \vec{E}_j folgt damit durch Überlagerung der beiden Teilfelder zu

$$\vec{E}_j = \vec{E}_+ + \vec{E}_- = \frac{q_{L,j}}{2\pi\epsilon_0\epsilon_r} \cdot \left(\frac{\vec{x} - \vec{x}_{L,j}}{|\vec{x} - \vec{x}_{L,j}|^2} - \frac{\vec{x} - \vec{x}_{Lg,j}}{|\vec{x} - \vec{x}_{Lg,j}|^2} \right) \quad (3.11)$$

$\vec{x}_{L,j}$ bezeichnet den Ortsvektor der Ersatzladung, $\vec{x}_{Lg,j}$ den Ortsvektor der Spiegelladung. Ferner gilt zu beachten, dass es sich bei der Ladung $q_{L,j}$ um eine Linienladungsdichte in C/m handelt.

Durch Integration des elektrischen Feldes parallel zur y-Achse gemäss

$$\phi(x,y) = -\int_{\gamma} \vec{E} \cdot d\vec{s} = -\int_0^y \vec{E} \cdot \begin{pmatrix} 0 \\ dy \end{pmatrix} \quad (3.12)$$

folgt für das elektrische Potential in Abhängigkeit des Aufpunktes \vec{x} folgende Gleichung:

$$\phi_j(\vec{x}) = \frac{q_{L,j}}{2\pi\epsilon_0\epsilon_r} \cdot \log \left(\frac{|\vec{x} - \vec{x}_{Lg,j}|}{|\vec{x} - \vec{x}_{L,j}|} \right) \quad (3.13)$$

Unter Berücksichtigung von Gleichung (3.4) lassen sich die Potentialkoeffizienten p_{ij} (Einträge der Matrix P_{12}) aus der Formel (3.13) durch Einsetzen des Ortsvektors des Konturpunktes $\vec{x}_{K,i}$ als Gewichtungsfaktor der Ladung $q_{L,j}$ gemäss Gleichung (3.14) beschreiben.

$$p_{ij} = \frac{1}{2\pi\epsilon_0\epsilon_r} \cdot \log \left(\frac{|\vec{x}_{K,i} - \vec{x}_{Lg,j}|}{|\vec{x}_{K,i} - \vec{x}_{L,j}|} \right) \quad (3.14)$$

3.4 Algorithmus zur Berechnung der elektrischen Randfeldstärken

Durch die Berücksichtigung elektrischer Systeme mit unterschiedlichen Frequenzen kann die Berechnung des elektrischen Feldes nicht mittels Phasoren erfolgen. Mit der Methode *Least Squares* könnten die zu verschiedenen Zeitpunkten quasistatisch gerechneten Feldstärken zur Bestimmung der Parameter eines geeigneten Ansatzes des zeitlichen Feldverlaufs benutzt werden. Dieses Verfahren stellte sich bei der Implementation jedoch als zu rechenintensiv heraus, sodass auch dieser Lösungsansatz verworfen wurde. Im Folgenden soll ein Algorithmus zur Berechnung der RMS-Randfeldstärken entwickelt werden, welcher analytisch motiviert ist und zur Laufzeit schnell Berechnungsergebnisse liefert. Die Implementation der Feldberechnung erfolgt in der Klasse *ElectricField* und ist in Abbildung 3.4 dargestellt.

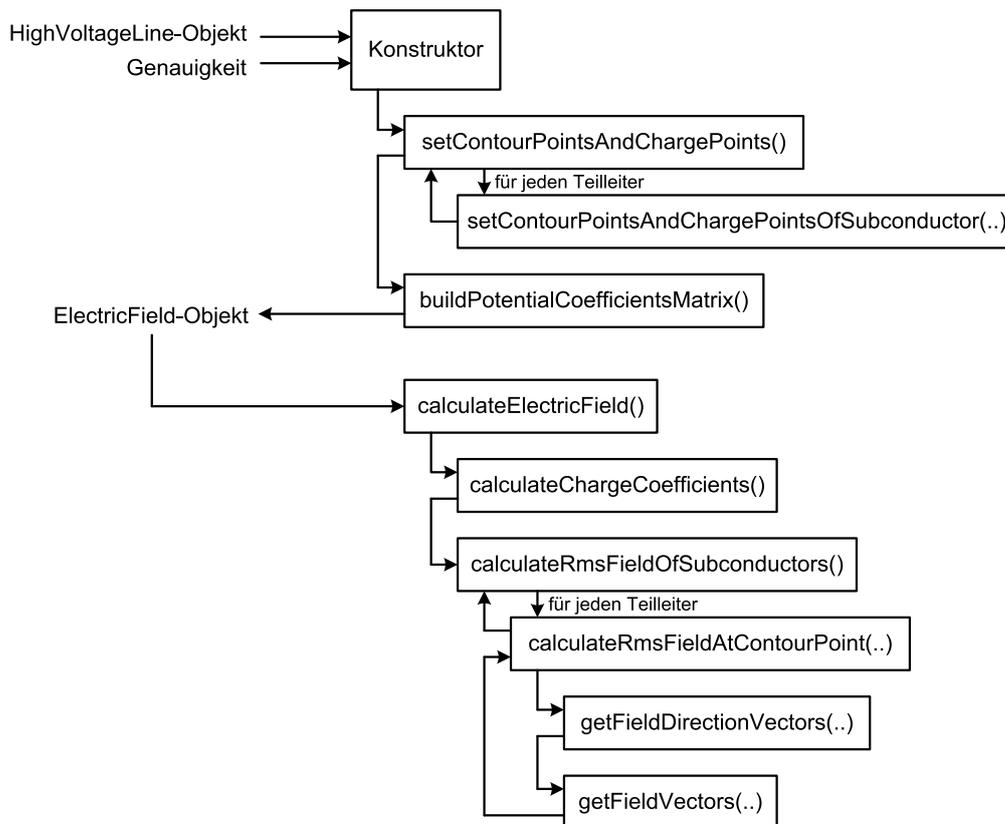


Abbildung 3.4: Flussdiagramm zur Implementation der Berechnung der elektrischen Randfeldstärken

3.4.1 Definition der Potentialverläufe

In einem ersten Schritt werden die möglichen zeitlichen Potentialverläufe definiert. Für Dreiphasenwechselstromsysteme sind dies gemäss (3.15), (3.16) und (3.17) drei sinusförmige um je $\frac{2\pi}{3}$ phasenverschobene Spannungsverläufe mit einer Kreisfrequenz von $\omega_1 = 2\pi \cdot 50$ Hz und

einem Amplitudenvektor bestehend aus Spaltenvektoren mit gleichen Einträgen für jedes System.

$$[\phi_{1,Ph1,u}] = \begin{pmatrix} [\hat{U}_{1,1}] \\ [\hat{U}_{1,2}] \\ \vdots \\ [\hat{U}_{1,\alpha}] \end{pmatrix} \cdot \sin(\omega_1 t) \quad (3.15)$$

$$[\phi_{1,Ph2,u}] = \begin{pmatrix} [\hat{U}_{1,1}] \\ [\hat{U}_{1,2}] \\ \vdots \\ [\hat{U}_{1,\alpha}] \end{pmatrix} \cdot \sin\left(\omega_1 t - \frac{2\pi}{3}\right) \quad (3.16)$$

$$[\phi_{1,Ph3,u}] = \begin{pmatrix} [\hat{U}_{1,1}] \\ [\hat{U}_{1,2}] \\ \vdots \\ [\hat{U}_{1,\alpha}] \end{pmatrix} \cdot \sin\left(\omega_1 t - \frac{4\pi}{3}\right) \quad (3.17)$$

Die durch eckige Klammern angedeuteten inneren Spaltenvektoren definieren jeweils unterschiedliche Dreiphasenwechselstromsysteme mit ihren entsprechenden Spannungsamplituden in V. Die Einträge innerhalb eines solchen Vektors beschreiben die Amplituden der Spannungsverläufe der einzelnen Konturpunkte innerhalb des Systems und sind damit gleich. u bezeichnet den Laufindex über die im jeweiligen Potentialvektor definierten Potentiale der Konturpunkte. Die Indizes $Ph1$, $Ph2$ und $Ph3$ stehen für die drei Phasen des Systems.

So entspricht der Eintrag $[\hat{U}_{1,i}]$ einem Vektor mit Einträgen der Amplitude der Phasenspannung des i -ten Dreiphasensystems. Die Anzahl Zeilen der inneren Vektoren richtet sich nach der Anzahl Konturpunkte, welche diesen Potentialverlauf besitzen.

Weiter wird nach (3.18) der Potentialverlauf des Einphasenwechselstromsystems als Sinusschwingung mit einer Kreisfrequenz von $\omega_2 = 2\pi \cdot 16.7$ Hz eingeführt. Es wird davon ausgegangen, dass zwischen der ersten Phase des Drehstromsystems und des Einphasensystems keine Phasenverschiebung auftritt.

$$[\phi_{2,Ph1,v}] = \begin{pmatrix} [\hat{U}_{2,1}] \\ [\hat{U}_{2,2}] \\ \vdots \\ [\hat{U}_{2,\beta}] \end{pmatrix} \cdot \sin(\omega_2 t) \quad (3.18)$$

3 Berechnung der elektrischen Randfeldstärken mittels Ersatzladungsverfahren

Die inneren Vektoren beschreiben in diesem Fall die Spannungsamplituden der Einphasenwechselstromsysteme. v bezeichnet den Laufindex über die im Potentialvektor definierten Potentiale der Konturpunkte.

Zuletzt werden die konstanten Potentiale für die Gleichstromsysteme gemäss (3.19) und (3.20) definiert. Die inneren Vektoren bilden hier die Gleichspannungen der entsprechenden Systeme. w bezeichnet den Laufindex über die im Potentialvektor definierten Potentiale der Konturpunkte. Die Indizes $Ph1$ und $Ph2$ stehen für die zwei Phasen des Systems.

$$[\phi_{3,Ph1,w}] = \begin{pmatrix} [U_{3,1}] \\ [U_{3,2}] \\ \vdots \\ [U_{3,\gamma}] \end{pmatrix} \quad (3.19)$$

$$[\phi_{3,Ph2,w}] = \begin{pmatrix} [U_{4,1}] \\ [U_{4,2}] \\ \vdots \\ [U_{4,\gamma}] \end{pmatrix} \quad (3.20)$$

Mit obigen Definitionen existieren also α Dreiphasen- und β Einphasenwechselstromsysteme. Die Anzahl der Gleichstromsysteme beträgt γ . Werden die Vektoren nun zu einem Potentialvektor $[\phi_{K,i}]$ zusammengefasst, können die zeitabhängigen Ersatzladungen nach Gleichung (3.5) bestimmt werden.

3.4.2 Berechnung der Ersatzladungen

Um schliesslich das elektrische Feld in den Konturpunkten der Leiterseilumrandung bestimmen zu können, müssen die Ersatzladungen unter Einbezug der entsprechenden Potentialverläufe berechnet werden. Sobald die zeitlichen Ladungsverläufe bekannt sind, kann durch Superposition der durch die Ersatzladungen erzeugten elektrischen Teilfelder die Gesamtfeldstärke ermittelt werden.

Platzierung der Konturpunkte und Ersatzladungen

Zunächst werden die Konturpunkte geordnet nach der Zeitabhängigkeit ihres Potentialverlaufs und die dazugehörigen Ersatzladungen platziert (siehe Abschnitt 3.2). Die positionierten Ersatzladungen besitzen somit dieselbe Reihenfolge wie die in Abschnitt 3.4.1 definierten Potentialverläufe.

Die Platzierung erfolgt in der Klasse *ElectricField* direkt beim Aufruf des Konstruktors über die Methode *setContourPointsAndChargePoints()*. Die Konturpunkte und Ersatzladungen werden über die innere Funktion *setContourPointsAndChargePointsOfSubconductor(..)* gemäss der Formel (3.8) positioniert und in unabhängigen *ArrayList*-Instanzen gespeichert.

Berechnung der Potentialkoeffizienten

Im nächsten Schritt werden die Potentialkoeffizienten über die Funktion *buildPotentialCoefficientsMatrix()*, welche ebenfalls im Konstruktor aufgerufen wird, berechnet. Für jeden Konturpunkt ist der Einfluss aller Ersatzladungen zu berücksichtigen. Zur Bestimmung der Potentialkoeffizientenmatrix findet eine äussere Iteration über die Konturpunkte (bzw. über die Zeilen der Matrix) sowie eine innere über die Ersatzladungen (bzw. über die Spalten der Matrix) statt. In der inneren Schleife berechnet sich der Potentialkoeffizient nach Gleichung (3.14), wobei der Zeilenindex i den Konturpunkt und der Spaltenindex j die Ersatzladung bezeichnet. Für die weitere Verwendung wird über das Java-Paket *JAMA* die Inverse der Potentialkoeffizientenmatrix P_{12}^{-1} gebildet und gespeichert.

Berechnung der Ladungskoeffizienten

Da die elektrischen Potentiale der Konturpunkte von der Zeit abhängen, können die Ersatzladungen nicht einfach durch Lösen von Gleichung (3.5) numerisch berechnet werden. Die Zeitabhängigkeit muss bei der Berechnung der Ladungen also gesondert mitgeführt werden. Wird die Kapazitätsmatrix $C = P_{12}^{-1}$ mit dem Vektor $[\phi_{K,i}]$, welcher die zeitlichen Potentialverläufe wie zuvor definiert enthält, multipliziert, folgt für die Ersatzladung folgende Gleichung:

$$q_{L,z} = A_z \cdot \sin(\omega_1 t) + B_z \cdot \sin\left(\omega_1 t - \frac{2\pi}{3}\right) + C_z \cdot \sin\left(\omega_1 t - \frac{4\pi}{3}\right) + D_z \cdot \sin(\omega_2 t) + E_z \quad (3.21)$$

Die Koeffizienten A_z , B_z , C_z , D_z und E_z stellen die Ladungskoeffizienten der z -ten Ersatzladung dar und werden gemäss den Gleichungen (3.22) bis (3.26) bestimmt. Die c_{ij} bezeichnen dabei die Einträge der Kapazitätsmatrix C . Die Anzahl der zu summierenden Kapazitätseinträge pro Spannungsfaktor bestimmt sich durch die Anzahl Konturpunkte, die den entsprechenden durch den Spannungsfaktor definierten Potentialverlauf zeigen.

$$A_z = (c_{z1} + c_{z2} + \dots + c_{za}) \cdot \widehat{U}_{1,1} + (c_{z(a+1)} + c_{z(a+2)} + \dots + c_{zb}) \cdot \widehat{U}_{1,2} \\ + \dots + (c_{zc} + c_{z(c+1)} + \dots + c_{zd}) \cdot \widehat{U}_{1,\alpha} \quad (3.22)$$

$$B_z = (c_{z(d+1)} + c_{z(d+2)} + \dots + c_{ze}) \cdot \widehat{U}_{1,1} + (c_{z(e+1)} + c_{z(e+2)} + \dots + c_{zf}) \cdot \widehat{U}_{1,2} \\ + \dots + (c_{zg} + c_{z(g+1)} + \dots + c_{zh}) \cdot \widehat{U}_{1,\alpha} \quad (3.23)$$

$$C_z = (c_{z(h+1)} + c_{z(h+2)} + \dots + c_{zi}) \cdot \widehat{U}_{1,1} + (c_{z(i+1)} + c_{z(i+2)} + \dots + c_{zj}) \cdot \widehat{U}_{1,2} \\ + \dots + (c_{zk} + c_{z(k+1)} + \dots + c_{zl}) \cdot \widehat{U}_{1,\alpha} \quad (3.24)$$

$$D_z = (c_{z(l+1)} + c_{z(l+2)} + \dots + c_{zm}) \cdot \widehat{U}_{2,1} + (c_{z(m+1)} + c_{z(m+2)} + \dots + c_{zn}) \cdot \widehat{U}_{2,2} + \dots + (c_{zo} + c_{z(o+1)} + \dots + c_{zp}) \cdot \widehat{U}_{2,\beta} \quad (3.25)$$

$$E_z = (c_{z(p+1)} + c_{z(p+2)} + \dots + c_{zq}) \cdot U_{3,1} + (c_{z(q+1)} + c_{z(q+2)} + \dots + c_{zr}) \cdot U_{3,2} + \dots + (c_{zs} + c_{z(s+1)} + \dots + c_{zt}) \cdot U_{3,\gamma} + (c_{z(t+1)} + c_{z(t+2)} + \dots + c_{zu}) \cdot U_{4,1} + (c_{z(u+1)} + c_{z(u+2)} + \dots + c_{zv}) \cdot U_{4,2} + \dots + (c_{zw} + c_{z(w+1)} + \dots + c_{zx}) \cdot U_{4,\gamma} \quad (3.26)$$

Zur Bestimmung der Ersatzladungen müssen also je fünf Ladungskoeffizienten berechnet werden. Dies erfolgt in der Methode `calculateChargeCoefficients()`, welche über die dem Benutzer zur Verfügung stehende Funktion `calculateElectricField()` aufgerufen wird.

Die Ladungskoeffizienten für jede Ladung werden innerhalb einer Iteration über die Ersatzladungen bestimmt. Für jeden Koeffizienten wird über die Anzahl der elektrischen Systeme mit entsprechender Zeitabhängigkeit iteriert. Die relevanten Kapazitätsmatrixeinträge werden dann pro System summiert und mit der Betriebsspannung multipliziert. Durch Addition dieser Berechnungsergebnisse am Ende jedes inneren Schleifendurchlaufs entsteht der gesuchte Ladungskoeffizient der durch die äussere Schleife definierten Ersatzladung.

Damit die Anzahl zu summierender Kapazitätsmatrixeinträge für jedes System bekannt ist, werden bei der Platzierung der Ladungen in `setContourPointsAndChargePoints()` die Variablen `numberOfPotentialCoefficientsK` für $K = A, B, C, D, E$ als Arrays eingeführt, deren Grösse durch die Anzahl relevanter Systeme bestimmt wird. Jedes Element definiert dann die Anzahl zu summierender Kapazitätsmatrixeinträge des elektrischen Systems.

3.4.3 Berechnung der RMS-Randfeldstärken

Nachdem nun die zeitabhängigen Ersatzladungen mittels ihrer Ladungskoeffizienten bestimmt worden sind, können die von ihnen erzeugten elektrischen Teilfelder vektoriell überlagert werden. Wiederum ist die Zeitabhängigkeit bei der Superposition in den Gleichungen gesondert zu betrachten.

Vorgehen

Das elektrische Feld im Aufpunkt \vec{x} ist unter Berücksichtigung von Gleichung (3.11) durch die Formel (3.27) gegeben. Dabei bezeichnet n die Anzahl Konturpunkte resp. Ersatzladungen der gesamten Hochspannungsfreileitung.

$$\begin{aligned}
 \vec{E}(\vec{x}) &= \left(A_1 \cdot \sin(\omega_1 t) + B_1 \cdot \sin\left(\omega_1 t - \frac{2\pi}{3}\right) + C_1 \cdot \sin\left(\omega_1 t - \frac{4\pi}{3}\right) + D_1 \cdot \sin(\omega_2 t) + E_1 \right) \\
 &\cdot \left(\frac{\vec{x} - \vec{x}_{L,1}}{|\vec{x} - \vec{x}_{L,1}|^2} - \frac{\vec{x} - \vec{x}_{Lg,1}}{|\vec{x} - \vec{x}_{Lg,1}|^2} \right) \cdot \frac{1}{2\pi\epsilon_0\epsilon_r} \\
 &+ \dots \\
 &+ \left(A_n \cdot \sin(\omega_1 t) + B_n \cdot \sin\left(\omega_1 t - \frac{2\pi}{3}\right) + C_n \cdot \sin\left(\omega_1 t - \frac{4\pi}{3}\right) + D_n \cdot \sin(\omega_2 t) + E_n \right) \\
 &\cdot \left(\frac{\vec{x} - \vec{x}_{L,n}}{|\vec{x} - \vec{x}_{L,n}|^2} - \frac{\vec{x} - \vec{x}_{Lg,n}}{|\vec{x} - \vec{x}_{Lg,n}|^2} \right) \cdot \frac{1}{2\pi\epsilon_0\epsilon_r} \tag{3.27}
 \end{aligned}$$

Durch Zusammenfassen von Termen gleicher Zeitabhängigkeit folgt aus Formel (3.27) folgende Gleichung:

$$\begin{aligned}
 \vec{E}(\vec{x}) &= \alpha \cdot (A_1 \vec{v}_1 + A_2 \vec{v}_2 + \dots + A_n \vec{v}_n) \cdot \sin(\omega_1 t) \\
 &+ \alpha \cdot (B_1 \vec{v}_1 + B_2 \vec{v}_2 + \dots + B_n \vec{v}_n) \cdot \sin\left(\omega_1 t - \frac{2\pi}{3}\right) \\
 &+ \alpha \cdot (C_1 \vec{v}_1 + C_2 \vec{v}_2 + \dots + C_n \vec{v}_n) \cdot \sin\left(\omega_1 t - \frac{4\pi}{3}\right) \\
 &+ \alpha \cdot (D_1 \vec{v}_1 + D_2 \vec{v}_2 + \dots + D_n \vec{v}_n) \cdot \sin(\omega_2 t) \\
 &+ \alpha \cdot (E_1 \vec{v}_1 + E_2 \vec{v}_2 + \dots + E_n \vec{v}_n) \tag{3.28}
 \end{aligned}$$

Dabei gilt für den Faktor α die Gleichung (3.29). Neu eingeführt wurden die Vektoren $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n$ gemäss Formel (3.30), welche unter anderem die Richtungen der Teilfelder enthalten.

$$\alpha = \frac{1}{2\pi\epsilon_0\epsilon_r} \tag{3.29}$$

$$\vec{v}_i = \left(\frac{\vec{x} - \vec{x}_{L,i}}{|\vec{x} - \vec{x}_{L,i}|^2} - \frac{\vec{x} - \vec{x}_{Lg,i}}{|\vec{x} - \vec{x}_{Lg,i}|^2} \right) \tag{3.30}$$

Die Beschreibung des elektrischen Feldes liegt nun als zeitabhängiger Vektor im Aufpunkt \vec{x} vor. Für die spätere Verwendung zur Berechnung der akustischen Geräuschemissionen sollen nun noch die RMS-Werte der Randfeldstärken bestimmt werden. Dazu werden nach Gleichungen (3.31) bis (3.35) die Feldvektoren $\vec{w}_1, \vec{w}_2, \vec{w}_3, \vec{w}_4$ und \vec{w}_5 eingeführt.

$$\vec{w}_1 = \alpha \cdot (A_1 \vec{v}_1 + A_2 \vec{v}_2 + \dots + A_n \vec{v}_n) \tag{3.31}$$

$$\vec{w}_2 = \alpha \cdot (B_1 \vec{v}_1 + B_2 \vec{v}_2 + \dots + B_n \vec{v}_n) \quad (3.32)$$

$$\vec{w}_3 = \alpha \cdot (C_1 \vec{v}_1 + C_2 \vec{v}_2 + \dots + C_n \vec{v}_n) \quad (3.33)$$

$$\vec{w}_4 = \alpha \cdot (D_1 \vec{v}_1 + D_2 \vec{v}_2 + \dots + D_n \vec{v}_n) \quad (3.34)$$

$$\vec{w}_5 = \alpha \cdot (E_1 \vec{v}_1 + E_2 \vec{v}_2 + \dots + E_n \vec{v}_n) \quad (3.35)$$

Die RMS-Werte der beiden Komponenten des Feldvektors \vec{E} lassen sich durch Integration der quadrierten zeitabhängigen Feldkomponente über eine Periode gemäss Gleichung (3.36) gewinnen. Dabei entspricht die Integrationsperiode der Periode einer Schwingung mit $16\frac{2}{3}$ Hz, das heisst auf 50 Hz bezogen sind dies 6π .

$$\begin{aligned} E_{rms,i} &= \sqrt{\frac{1}{T} \int_0^T E_i^2(t) dt} \\ &= \sqrt{\frac{1}{6\pi} \int_0^{6\pi} \left(w_{1,i} \sin(x) + w_{2,i} \sin\left(x - \frac{2\pi}{3}\right) + w_{3,i} \sin\left(x - \frac{4\pi}{3}\right) + w_{4,i} \sin\left(\frac{x}{3}\right) + w_{5,i} \right)^2 dx} \\ &= \sqrt{\frac{1}{2} \left(w_{1,i}^2 - w_{1,i}(w_{2,i} + w_{3,i}) + w_{2,i}^2 - w_{2,i}w_{3,i} + w_{3,i}^2 + w_{4,i}^2 + 2w_{5,i}^2 \right)} \end{aligned} \quad (3.36)$$

Der Index i steht für die vektoriellen Komponenten in x- und y-Richtung ($i = 1$ bzw. $i = 2$).

Schlussendlich definiert sich der RMS-Wert der elektrischen Feldstärke als Betrag des RMS-Feldvektors gemäss Gleichung (3.37). Hierbei ist zu beachten, dass die Abhängigkeit vom Ortspunkt \vec{x} , wo das Feld berechnet werden soll, bereits bei der Definition der Vektoren \vec{v}_i gemäss Gleichung (3.30) in die Formeln einfließt. In den Folgegleichungen wurde diese Abhängigkeit nicht mehr explizit angegeben.

$$E_{rms} = \sqrt{E_{rms,1}^2 + E_{rms,2}^2} \quad (3.37)$$

Implementation

Wie die Berechnung der Ladungskoeffizienten wird auch die Feldberechnung (Funktion `calculateRmsFieldOfSubconductors()`) über die dem Benutzer zur Verfügung stehende Funktion `calculateElectricField()` aufgerufen. Die Bestimmung der RMS-Randfeldstärken geschieht durch Iteration über die Gesamtanzahl der Teileiter der Hochspannungsfreileitung. Der eigentliche

Rechenalgorithmus wird für jeden Konturpunkt des Teilleiters durch Aufruf der inneren Methode *calculateRmsFieldAtContourPoint(..)* ausgeführt. Für jeden Teilleiter werden die RMS-Feldstärken an den Konturpunkten in einem *Array* gespeichert, welches dann als Eintrag einer *ArrayList*-Instanz abgelegt wird. So können die Feldwerte eines Teilleiters später einfach abgefragt werden.

Im Innern der Funktion *calculateRmsFieldAtContourPoint(..)* werden über die Methode *getFieldDirectionVectors(..)* die Vektoren $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n$ gemäss Gleichung (3.30) erhalten. Dies erfolgt durch Iteration über die Anzahl Ersatzladungen n . Der Aufpunkt wird durch den übergebenen Konturpunktindex spezifiziert.

Anschliessend folgt die Berechnung der Feldvektoren \vec{w}_1 bis \vec{w}_5 in der Methode *getFieldVectors(..)* nach den Formeln (3.31) bis (3.35). Wiederum erfolgt eine Iteration über die Anzahl Ersatzladungen, wobei in jedem Schritt i für jeden Feldvektor der Vektor \vec{v}_i mit dem entsprechenden Ladungskoeffizienten K_i mit $K = A, B, C, D, E$ multipliziert wird. Diese Ergebnisse werden dann über die Schleife aufsummiert und am Schluss mit dem Faktor α gemäss Gleichung (3.29) korrigiert.

Sobald die Feldvektoren vorliegen, kann in *calculateRmsFieldAtContourPoint(..)* die Berechnung der RMS-Feldkomponenten nach Gleichung (3.36) durchgeführt und darauffolgend der RMS-Wert der elektrischen Randfeldstärke gemäss Formel (3.37) bestimmt werden.

Funktionen der Klasse *ElectricField*

Bei der Erstellung einer Instanz der Klasse *ElectricField* werden durch Übergabe des *HighVoltageLine*-Objekts sowie der gewünschten Rechengenauigkeit (siehe Abschnitt 3.5) direkt die Konturpunkte und Ersatzladungen platziert und daraufhin die Kapazitätsmatrix C berechnet. Diese zwei Aufgaben müssen also nicht manuell gestartet werden. Die Klasse *ElectricField* stellt folgende Funktionen zur Verfügung:

- **void calculateElectricField()**
Berechnet die RMS-Werte der elektrischen Randfeldstärken in den Konturpunkten aller Teilleiter. Dazu werden *calculateChargeCoefficients()* und anschliessend *calculateRmsFieldOfSubconductors()* aufgerufen.
- **double getMaximumRmsFieldOfSubconductor(int subconductorIndex)**
Gibt die maximale RMS-Randfeldstärke des durch den Index (Ordnung nach Potentialverläufen) spezifizierten Teilleiters zurück.
- **double getAverageRmsFieldOfSubconductor(int subconductorIndex)**
Gibt den Mittelwert der RMS-Randfeldstärken des durch den Index (Ordnung nach Potentialverläufen) spezifizierten Teilleiters über die Teilleiterkontur zurück.
- **double[] getRmsFieldOfSubconductor(int systemIndex, int conductorIndex, int subconductorIndex)**
Liefert alle RMS-Randfeldstärken in den zum spezifizierten Teilleiter gehörenden Konturpunkten als *Array*.

3.5 Genauigkeit

Nachdem die Berechnung der elektrischen Randfeldstärken implementiert ist, stellt sich im Folgenden die Frage, wieviele Konturpunkte pro Teilleiter nötig sind, um eine maximale Randfeldstärke angemessener Genauigkeit zu erhalten. Hierbei ist zu beachten, dass durch die Anzahl Konturpunkte eine Schrittweite über den Umfangswinkel definiert wird, welche direkten Einfluss auf die Genauigkeit der Maximalfeldstärke ausübt. Dieser Zusammenhang soll im Folgenden analytisch hergeleitet werden. Dabei wird von einem ideal sinusförmigen Verlauf der Randfeldstärke über der Teilleiterkontur ausgegangen. Dies stellt jedoch nur eine Näherung dar. Durch die endliche Anzahl Ersatzladungen zeigt die Randfeldstärke stets eine Schwankung um ihren mittleren sinusförmigen Verlauf, welche mit zunehmender Anzahl Ersatzladungen abgeschwächt wird.

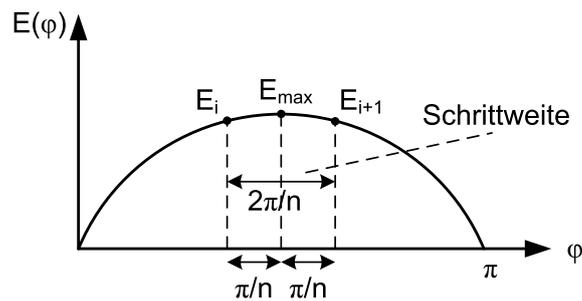


Abbildung 3.5: Skizze zur Bestimmung des maximalen Feldstärkefehlers in Abhängigkeit der Anzahl Konturpunkte

Aus Abbildung 3.5 ist ersichtlich, dass die maximale Feldabweichung der um die Maximalfeldstärke liegenden Konturpunkte genau dann am grössten ist, wenn diese die Hälfte der Schrittweite vom Maximum entfernt liegen. Aus dieser Feststellung folgt für die maximale Feldabweichung ΔE folgende Gleichung:

$$\Delta E \leq E_{max} - E_{max} \cdot \sin\left(\frac{\pi}{2} + \frac{\pi}{n}\right) \quad (3.38)$$

Wird nun durch E_{max} dividiert und die Sinusfunktion durch die entsprechende Kosinusfunktion ersetzt, folgt der maximale relative Fehler:

$$\Delta E_r \leq 1 - \cos\left(\frac{\pi}{n}\right) \quad (3.39)$$

Schlussendlich definiert Gleichung (3.40) die Mindestanzahl an Konturpunkten.

$$n \geq \frac{\pi}{\arccos(1 - \Delta E_r)} \quad (3.40)$$

Die gewünschte Rechengenauigkeit ΔE_r wird wie bereits erwähnt dem Konstruktor der Klasse *ElectricField* übergeben. Vom Benutzer können die in Tabelle 3.1 angegebenen Genauigkeiten

über die grafische Oberfläche eingestellt werden. Ebenfalls ist die durch Formel (3.40) resultierende Anzahl Konturpunkte angegeben.

Genauigkeit ΔE_r	Konturpunkte
0.04	11
0.02	16
0.01	22
0.005	31
0.0025	44

Tabelle 3.1: Dem Benutzer zur Verfügung stehende Rechengenauigkeiten mit entsprechender Anzahl Konturpunkte

3.6 Vergleichsrechnung mit *Cmsol*

Um die implementierte Feldberechnung und damit die Überlagerung der Feldanteile unterschiedlicher Systeme hinsichtlich der Rechenergebnisse beurteilen zu können, wird die fiktive Hochspannungsfreileitung aus Abbildung 3.6 in *HVLNoise* sowie *Cmsol* berechnet und die Ergebnisse verglichen. Die Hochspannungsfreileitung besteht aus einem Dreiphasenwechsel-, Einphasenwechsel- sowie einem Gleichstromsystem. Insgesamt resultieren also sechs Leiter. Die Teilleitergeometrie definiert sich durch zwei im Abstand von 400 mm angeordneten Teilleitern mit je einem Durchmesser von 30 mm.

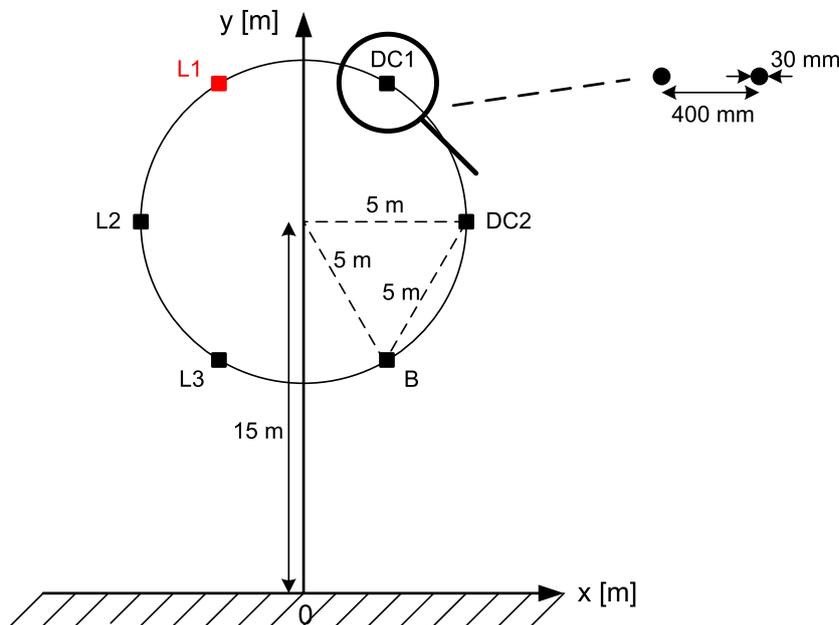


Abbildung 3.6: Fiktive Leitungsgeometrie zum Vergleich der Feldberechnung mit *Cmsol*; L1, L2, L3 Phasen des Drehstromsystems; B Bahnstromsystem; DC1, DC2 Gleichstromsystem

3 Berechnung der elektrischen Randfeldstärken mittels Ersatzladungsverfahren

Die x- und y-Koordinaten der Leiter sowie Teilleiter sind in Tabelle 3.2 zusammengefasst.

Leiter	Mittelpunkt	Teilleiter links	Teilleiter rechts
DC2	(5, 15)	(4.8, 15)	(5.2, 15)
DC1	(2.5, 19.33)	(2.3, 19.33)	(2.7, 19.33)
L1	(-2.5, 19.33)	(-2.7, 19.33)	(-2.3, 19.33)
L2	(-5, 15)	(-5.2, 15)	(-4.8, 15)
L3	(-2.5, 10.67)	(-2.7, 10.67)	(-2.3, 10.67)
B	(2.5, 10.67)	(2.3, 10.67)	(2.7, 10.67)

Tabelle 3.2: Koordinaten der Leiter sowie Teilleiter in m der Mastgeometrie zum Vergleich der Feldberechnung mit *Comsol*

Die Betriebsspannungen werden für das Gleichstromsystem zu ± 310 kV, für das Dreiphasenwechselstromsystem zu 380 kV und für das Einphasenwechselstromsystem zu 220 kV gewählt. Die Feldberechnung erfolgt zum Zeitpunkt $t = 15$ ms. Gemessen wird der Randfeldstärkeverlauf des linken Teilleiters des Leiters L1. Die Rechenergebnisse werden in Abbildung 3.7 verglichen.

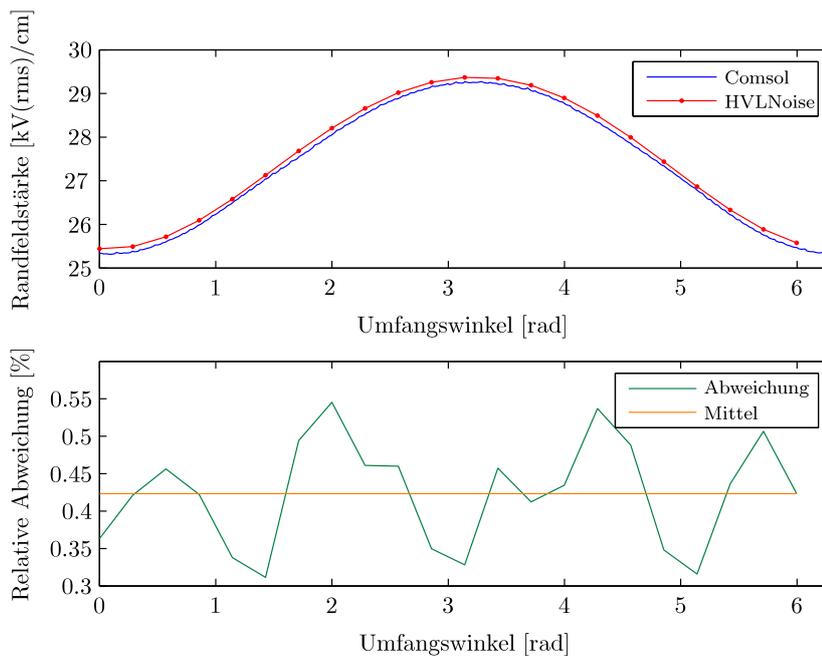


Abbildung 3.7: Vergleich der Randfeldstärke des linken Teilleiters von L1 mit *Comsol* und relativer Fehler

Die Vergleichsrechnung mit *Comsol* zeigt deutlich, dass das implementierte Ersatzladungsverfahren gute Rechenresultate liefert. Allgemein liegen die elektrischen Feldstärken von *HVL-Noise* leicht über denjenigen von *Comsol*. Der relative Fehler liegt über den Teilleiterumfang in jedem Konturpunkt unter 0.55%. Die mittlere Feldabweichung zu *Comsol* beträgt lediglich

0.4232%. Die in *HVLNoise* implementierte Feldberechnung ist also hinsichtlich der Rechenergebnisse mit komplexen Feldberechnungsprogrammen vergleichbar.

3.7 Schlussfolgerungen

Mit der erfolgreichen Implementation der Feldberechnung mittels Ersatzladungsverfahren können nun die Randfeldstärken der einzelnen Teilleiter berechnet werden, welche im Folgenden als Grundlage zur Bestimmung der akustischen Geräuschemissionen dienen.

Der Vergleich der Feldberechnung mit *Comsol* hat gezeigt, dass der entwickelte Algorithmus mit komplexen Feldberechnungsprogrammen mithalten kann und somit gute Rechenergebnisse liefert. Auch der Rechenaufwand liegt für Mastgeometrien mit zwei Dreiphasenwechselstromsystemen mit Zweierbündelgeometrie für eine Genauigkeit von $\Delta E_r = 0.01$ im Bereich von nur wenigen Sekunden.

Zu beachten gilt, dass die gezeigte Implementation lediglich zur Bestimmung der Randfeldstärken dient. Für eine Evaluierung der Feldstärken im Raum um die Leiter sei auf eine Berechnung mit finiten Elementen verwiesen.

4 Berechnung der akustischen Geräuschemissionen

Nachdem nun die elektrischen Randfeldstärken jedes Teilleiters bekannt sind, werden im letzten Schritt die akustischen Geräuschemissionen der Hochspannungsfreileitung berechnet. Die zur Vorhersage der Emission im Folgenden aufgeführten Formeln stützen sich im Wesentlichen auf das Formelwerk des Electric Power Research Institute (EPRI) [2].

4.1 Berechnung der Schallpegel

Zur Ermittlung eines äquivalenten Dauerschallpegels L_{eq} der Hochspannungsfreileitung werden zunächst die L_5 - und L_{50} -Schallpegel berechnet. Diese beiden Ergebnisse fließen dann in die Bestimmung des L_{eq} -Schallpegels ein.

4.1.1 Schallpegel L_5

Der A-gewichtete Level L_5 beschreibt den Schallpegel, welcher zu 5% der Zeit während Regen überschritten wird. Diese Schallerzeugung wird gemäss EPRI auch als *Heavy-Rain Generation* bezeichnet. Für die Ableitung der folgenden Formeln gilt eine Regenrate von 6.5 mm/h.

Die A-gewichtete Schalleistung A in W/m bei starkem Regen definiert sich für einen Leiter mit n Teilleitern vom Durchmesser d (in cm) gemäss folgender Gleichung:

$$A = n^2 \cdot \left(\frac{d}{3.8} \right)^{4.4} \cdot A_1 \cdot K_n \quad (4.1)$$

Der Faktor K_n wird mit

$$K_n = \begin{cases} 5.6 & n = 1 \\ 1.8 & n = 2 \\ 1 & n \geq 3 \end{cases} \quad (4.2)$$

angegeben. A_1 entspricht der Schalleistung (in W/m), welche durch ein Leiterseil mit Durchmesser 3.8 cm erzeugt wird und gemäss

$$A_1 \text{ dB} = -13.6 - \frac{665}{E} \quad (4.3)$$

4 Berechnung der akustischen Geräuschemissionen

von der mittleren maximalen Randfeldstärke E der im Leiterbündel vorhandenen Teilleiter abhängt.

Im Sinne einer Darstellung der einzelnen Teilleiter als inkohärente Schallquellen wird im Folgenden von der Schalleistung eines Teilleiters gemäss

$$A = n \cdot \left(\frac{d}{3.8} \right)^{4.4} \cdot A_1 \cdot K_n \quad (4.4)$$

ausgegangen. Die Schalleistungen der einzelnen Teilleiter bilden durch Addition dann wiederum die Leistung des Bündels. Diese Darstellung entspricht den Formeln nach EPRI für den Teilleiter-Standardmodus. Mit den heutigen Angaben zum Formelwerk lässt sich für die beliebige Platzierung der Teilleiter (Expertenmodus) diese Formel nicht physikalisch herleiten.

Wird nun Gleichung (4.4) auf beiden Seiten logarithmiert und mit dem Faktor 10 multipliziert, folgt für die Schalleistung eines Teilleiters in Dezibel über 1 W/m unter Berücksichtigung von Gleichung (4.3) die Formel (4.5). E_{max} bezeichnet nun die maximale Randfeldstärke des Teilleiters.

$$A \text{ dB} = 10 \cdot \log_{10}(n) + 44 \cdot \log_{10}(d) - 39.1 - \frac{665}{E_{max}} + 10 \cdot \log_{10}(K_n) \quad (4.5)$$

Für $n \geq 3$ ist zudem gemäss EPRI das Verhältnis von maximaler zu mittlerer Randfeldstärke (k -Faktor) mittels Gleichung (4.6) additiv zu berücksichtigen.

$$K \text{ dB} = 22.9 \cdot \frac{E_{max}}{E_{avg}} - 30.2 \quad (4.6)$$

Schliesslich folgt für die Schalleistung eines Teilleiters in Dezibel über 1 W/m die Gleichung (4.7). Zu beachten gilt, dass die Orientierung der Teilleiter für ein Zweierbündel in der Formel nicht berücksichtigt wurde.

$$A \text{ dB} = \begin{cases} 44 \cdot \log_{10}(d) - 31.6 - 665/E_{max} & n = 1 \\ 10 \cdot \log_{10}(n) + 44 \cdot \log_{10}(d) - 36.5 - 665/E_{max} & n = 2 \\ 10 \cdot \log_{10}(n) + 44 \cdot \log_{10}(d) - 69.3 - 665/E_{max} + 22.9 \cdot E_{max}/E_{avg} & n \geq 3 \end{cases} \quad (4.7)$$

Um nun den A-gewichteten L_5 -Schallpegel der Geräuschemission der gesamten Hochspannungsfreileitung zu bestimmen, wird der Schallpegel eines Teilleiters $L_{5,i}$ in Dezibel über $20 \mu\text{Pa}$ nach Gleichung (4.8) berechnet. R_i bezeichnet dabei den Abstand des Teilleiters i vom Messpunkt in m. Die resultierenden Pegel werden dann gemäss Formel (4.9) als inkohärente Quellen überlagert.

$$L_{5,i} \text{ dB} = A \text{ dB} + 114.3 - 10 \cdot \log_{10}(R_i) - 0.02R_i \quad (4.8)$$

$$L_5 \text{ dB} = 10 \cdot \log_{10} \left(\sum_{i=1}^n 10^{L_{5,i} \text{ dB}/10} \right) \quad (4.9)$$

Der Term $-0.02R_i$ in Gleichung (4.8) berücksichtigt die Absorption der Schallenergie während sich die Schallwelle durch die Luft bewegt.

4.1.2 Schallpegel L_{50}

Beim A-gewichteten Level L_{50} handelt es sich um den Schallpegel, welcher zu 50% der Zeit während Regen überschritten wird. Diese Schallerzeugung wird gemäss EPRI auch als *Wet-Conductor Generation* bezeichnet. Für die Ableitung der folgenden Formeln gilt eine Regenrate von 0.75 mm/h.

Die Berechnung des L_{50} -Schallpegels basiert auf einer Korrektur des zuvor bestimmten L_5 -Levels. Zunächst wird die elektrische Randfeldstärke E_c eruiert, bei welcher der L_{50} -Schallpegel sechs Dezibel unterhalb des L_5 -Pegels liegt. Für einen Teilleiterdurchmesser d zwischen 2 cm und 8 cm ist der 6 dB-Gradient in Abhängigkeit der Anzahl Teilleiter n folgendermassen definiert:

$$E_c = \begin{cases} 24.4/d^{0.24} & n \leq 8 \\ 24.4/d^{0.24} - 0.25(n-8) & n > 8 \end{cases} \quad (4.10)$$

Durch Einführung von Gleichung (4.11) kann der Korrekturterm ΔA_{wc} in Abhängigkeit des zuvor definierten 6 dB-Gradienten berechnet werden.

$$\Delta A_{wc} \text{ dB} = \begin{cases} 8.2 - 14.2 \cdot E_c/E_{max} & n < 3 \\ 2.4 - 14.2 \cdot E_c/E_{max} + 8 \cdot E_{max}/E_{avg} & n \geq 3 \end{cases} \quad (4.11)$$

Für $n \geq 3$ findet wie schon bei der Definition der Schalleistung eines Teilleiters eine additive Anpassung mit dem k -Faktor statt. Der Schallpegel eines Teilleiters $L_{50,i}$ in Dezibel über 20 μPa wird nach Gleichung (4.12) unter Berücksichtigung von Formel (4.8) berechnet.

$$L_{50,i} \text{ dB} = L_{5,i} \text{ dB} + \Delta A_{wc} \text{ dB} \quad (4.12)$$

Der Gesamtpegel L_{50} bestimmt sich dann gemäss Gleichung (4.13) wie beim L_5 -Schallpegel.

$$L_{50} \text{ dB} = 10 \cdot \log_{10} \left(\sum_{i=1}^n 10^{L_{50,i} \text{ dB}/10} \right) \quad (4.13)$$

4.1.3 Energieäquivalenter Schallpegel L_{eq}

Nachdem nun die Berechnungen der L_5 - sowie L_{50} -Schallpegel beschrieben sind, kann die Bestimmung eines äquivalenten Dauerschallpegels erfolgen. Es sollen dabei zwei Fälle betrachtet werden. Zum einen der L_{eq} -Schallpegel, welcher bei Regen auftritt und zum anderen derjenige, der über eine Periode mit einer entsprechenden Regendauer resultiert.

Für den äquivalenten Dauerschallpegel $L_{eq,r}$ während Regen gilt gemäss [1] Gleichung (4.14) mit der Standardabweichung σ definiert nach Formel (4.15).

$$L_{eq,r} \text{ dB} = L_{50} \text{ dB} + \frac{0.115}{\text{dB}} \sigma^2 \quad (4.14)$$

$$\sigma = \frac{L_5 \text{ dB} - L_{50} \text{ dB}}{1.64} \quad (4.15)$$

Um daraus den $L_{eq,p}$ -Schallpegel über eine Periode mit relativer Regendauer d zu bestimmen, muss das zeitabhängige Quadrat des Schalldruckeffektivwertes bezogen auf den Referenzdruck im Quadrat über die gewünschte Periode T integriert werden:

$$\begin{aligned} L_{eq,p} \text{ dB} &= 10 \cdot \log_{10} \left(\frac{1}{T} \int_0^T \frac{p_{rms}^2(t)}{p_0^2} dt \right) \\ &= 10 \cdot \log_{10} \left(\int_0^d 10^{L_{eq,r} \text{ dB}/10} dt \right) \end{aligned} \quad (4.16)$$

Durch Ausrechnen folgt schliesslich für den Äquivalenzpegel über eine Periode mit entsprechender relativer Regendauer d die folgende Gleichung:

$$L_{eq,p} \text{ dB} = 10 \cdot \log_{10} \left(d \cdot 10^{L_{eq,r} \text{ dB}/10} \right) \quad (4.17)$$

4.1.4 Berücksichtigung der Regenrate

Die Berechnungsformeln nach EPRI beziehen sich für den L_5 -Schallpegel auf eine Regenrate von 6.5 mm/h, für den L_{50} -Schallpegel auf 0.75 mm/h. Für abweichende Regenraten wird die Korrektur des Schallpegels in Abhängigkeit der Regenrate in [2] als Grafik angegeben. Im Folgenden werden diese Kurven durch rationale Polynome mit Hilfe der *Curve Fitting Toolbox* von *MATLAB* angenähert. Abbildung 4.1 zeigt das Ergebnis. Die rote und blaue Kurve stellen die Polynome dar.

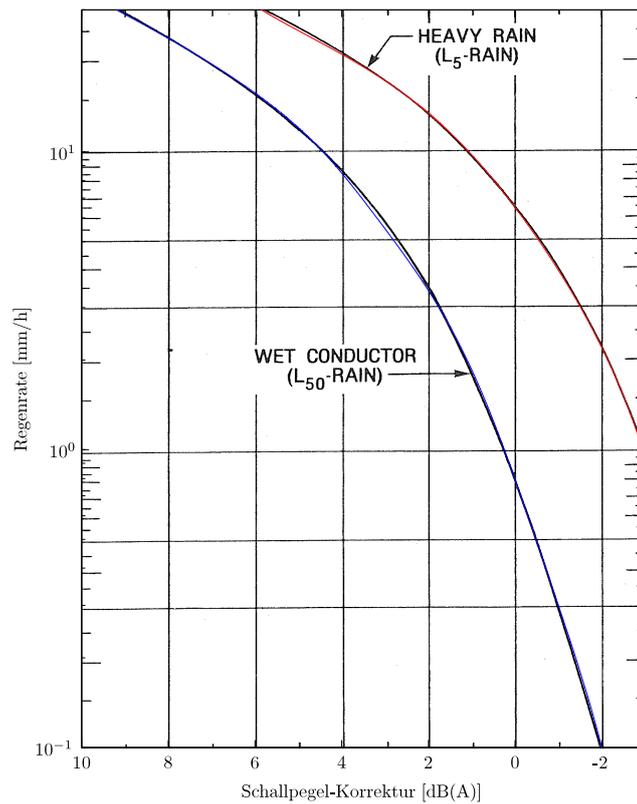


Abbildung 4.1: Kurvenfit zur Korrektur der Schallpegel zur Berücksichtigung der Regenrate

Die beiden Korrekturkurven werden jeweils durch rationale Polynome mit einem Zähler- und Nennergrad von vier nach Gleichungen (4.18) sowie (4.19) in Abhängigkeit der Regenrate x in mm/h beschrieben.

$$\Delta L_5 \text{ dB} = \frac{p_1 x^4 + p_2 x^3 + p_3 x^2 + p_4 x + p_5}{x^4 + q_1 x^3 + q_2 x^2 + q_3 x + q_4} \quad (4.18)$$

$$\Delta L_{50} \text{ dB} = \frac{r_1 x^4 + r_2 x^3 + r_3 x^2 + r_4 x + r_5}{x^4 + s_1 x^3 + s_2 x^2 + s_3 x + s_4} \quad (4.19)$$

Die Koeffizienten der Polynome sind in den Tabellen 4.1 sowie 4.2 angegeben.

4 Berechnung der akustischen Geräuschemissionen

Koeffizient	Wert
p_1	4.367
p_2	-84.13
p_3	4683.0
p_4	$-2.74 \cdot 10^4$
p_5	-4714.0
q_1	-40.59
q_2	952.5
q_3	7425.0
q_4	1005.0

Tabelle 4.1: Koeffizienten zum Fit des rationalen Polynoms (4.18) zur Schallpegelkorrektur gemäss Abbildung 4.1

Koeffizient	Wert
r_1	23.4
r_2	-233.5
r_3	1597.0
r_4	1509.0
r_5	-2097.0
s_1	35.58
s_2	-356.4
s_3	2846.0
s_4	684.1

Tabelle 4.2: Koeffizienten zum Fit des rationalen Polynoms (4.19) zur Schallpegelkorrektur gemäss Abbildung 4.1

Für die Schallpegel L_5 sowie L_{50} eines Teilleiters zu einer gegebenen Regenrate folgen dann unter Berücksichtigung der Korrekturterme die Gleichungen (4.20) und (4.21).

$$L_{5,i} \text{ dB} = L_{5,i} \text{ dB} + \Delta L_5 \text{ dB} \quad (4.20)$$

$$L_{50,i} \text{ dB} = L_{50,i} \text{ dB} + \Delta L_{50} \text{ dB} \quad (4.21)$$

Die anschliessende Überlagerung der Teilleiter als inkohärente Schallquellen geschieht weiterhin nach Gleichungen (4.9) sowie (4.13).

4.2 Benutzeroberfläche

Für den Benutzer erfolgt eine Beurteilung der akustischen Geräuschemissionen einer Hochspannungsfreileitung am einfachsten anhand eines äquivalenten Dauerschallpegels L_{eq} . Die statistischen Pegel L_5 sowie L_{50} sind dabei weniger aussagekräftig. In diesem Sinne sollen dem User zwei Berechnungsmodi zur Verfügung stehen:

- Berechnung des L_{eq} während Regen (Eingabe der Regenraten R_5 sowie R_{50}) sowie über eine Periode mit prozentualer Regendauer.
- Berechnung des L_{eq} zu einer gegebenen Regenrate.

Im ersten Modus können die Regenraten R_5 sowie R_{50} einzeln spezifiziert werden. Diese geben jeweils denjenigen Wert in mm/h an, welcher zu 5% resp. 50% der Zeit überschritten wird. Mit diesen beiden Eingaben werden dann der L_5 - sowie L_{50} -Pegel berechnet, um schliesslich auf den L_{eq} bei Regen zu schliessen. Darüber hinaus wird eine prozentuale Regendauer über eine bestimmte Periode (dies kann zum Beispiel ein Jahr sein) angegeben. Somit kann ein Äquivalenzpegel L_{eq} über den gewählten Zeitraum mit Regenspauzen ausgegeben werden.

Der zweite Modus erlaubt es dem Benutzer den L_{eq} -Schallpegel zu einer spezifizierten Regenrate auszurechnen. Somit können die akustischen Geräuschemissionen der Hochspannungsfreileitung bei variierender Regenintensität untersucht werden. Auch in diesem Fall werden zunächst der L_5 - und L_{50} -Schallpegel zur eingegebenen Regenrate berechnet, um dann den L_{eq} -Pegel bestimmen zu können.

Die Umsetzung der beiden wählbaren Modi in der Benutzeroberfläche ist in Abbildung 4.2 ersichtlich.

Abbildung 4.2: Grafische Oberfläche zur Wahl des Berechnungsmodus

4.3 Implementierung der Berechnung der akustischen Geräuschemissionen

Die Implementation zur Berechnung der Schallpegel erfolgt in der Klasse *AudibleNoise*. Zunächst soll aufbauend auf dem bisher Beschriebenen ein Vorgehen entwickelt werden, welches

4 Berechnung der akustischen Geräuschemissionen

die Distanzabhängigkeit der Schallpegel gesondert betrachtet.

Vorgehen

Um während dem Berechnungsvorgang möglichst viel der rechenintensiven Operationen ausführen zu können, wird Gleichung (4.9) resp. (4.13) so umgeformt, dass die Distanzabhängigkeit zum Messpunkt ausgeklammert werden kann. Es folgt damit Gleichung (4.22) mit $L_{5,i0}$ dB und $L_{50,i0}$ dB für den i -ten Teilleiter gemäss den Formeln (4.23) und (4.24).

$$L_p \text{ dB} = 10 \cdot \log_{10} \left(\left(10^{L_{p,10} \text{ dB}/10} + 10^{L_{p,20} \text{ dB}/10} + \dots \right) \cdot R_1^{-1} \cdot 10^{-0.002R_1} + \dots + \left(10^{L_{p,z0} \text{ dB}/10} + 10^{L_{p,(z+1)0} \text{ dB}/10} + \dots \right) \cdot R_k^{-1} \cdot 10^{-0.002R_k} \right) \quad (4.22)$$

p nimmt dabei je nach Schallpegel den Wert 5 oder 50 an. k spezifiziert dabei die Anzahl Leiter. Vor der Gewichtung mit der Distanzabhängigkeit des Leiters werden die Quadrate der Schalldruckeffektivwerte bezogen auf den Referenzdruck im Quadrat aller im Leiter vorhandenen Teilleiter aufsummiert.

$$L_{5,i0} \text{ dB} = A \text{ dB} + 114.3 + \Delta L_5 \text{ dB} \quad (4.23)$$

$$L_{50,i0} \text{ dB} = A \text{ dB} + 114.3 + \Delta A_{wc} \text{ dB} + \Delta L_{50} \text{ dB} \quad (4.24)$$

Die Distanz zwischen Leiter j und Messpunkt \vec{x} der Geräuschemission definiert sich gemäss Formel (4.25), wobei $\vec{x}_{M,j}$ die Position des Leitermittelpunktes angibt. Die Distanzen zwischen Leitermittelpunkt und tatsächlichem Mittelpunkt des Teilleiters werden vernachlässigt.

$$R_j = |\vec{x} - \vec{x}_{M,j}| \quad (4.25)$$

In einem nächsten Schritt werden nun Geräuschkoeffizienten $C_{Lp,j}$ für jeden Leiter jeweils für den L_5 - sowie L_{50} -Schallpegel ($p = 5$ bzw. $p = 50$) gemäss Gleichung (4.26) definiert, deren Berechnung vor Kenntnis der Messpunkte ausgeführt werden kann.

$$C_{Lp,j} = 10^{L_{p,m0} \text{ dB}/10} + 10^{L_{p,(m+1)0} \text{ dB}/10} + \dots \quad j = 1, 2, \dots, k \quad (4.26)$$

m gibt den ersten Teilleiterindex innerhalb des Leiters an. Über den Geräuschkoeffizienten werden alle Teilleiter innerhalb des Leiters j berücksichtigt.

Implementation

Mit dem Instanzieren eines Objekts der Klasse *AudibleNoise* werden dem Konstruktor das *HighVoltageLine*- sowie das durch die Feldberechnung gewonnene *ElectricField*-Objekt übergeben. Die Berechnung der Geräuschkoeffizienten für die beiden Level L_5 und L_{50} nach Gleichung (4.26) erfolgt dann über die dem Benutzer zur Verfügung stehende Funktion *calculateAudibleNoise(..)* durch Iteration über die Anzahl Leiter sowie eine innere Schleife über die Anzahl Teilleiter pro Leiter. Die Argumente sind die Regenraten R_5 sowie R_{50} und die prozentuale Regendauer. Um die richtigen Feldwerte gemäss der Ordnung der Leiter nach dem zeitabhängigen Potentialverlauf zu erhalten, geschieht die Iteration über die Leiter wie bei der Platzierung der Konturpunkte resp. Ersatzladungen. Die eigentliche Berechnung des Teilleiteranteils am Geräuschkoeffizienten erfolgt in der inneren Funktion *getSoundLevelsL5AndL50OfSubconductor(..)* gemäss den Gleichungen (4.23) und (4.24). Die übergebenen Argumente sind dabei der Teilleiterindex bezogen auf die Ordnung nach den Potentialverläufen, die Anzahl Teilleiter im Leiterbündel, der Teilleiterdurchmesser sowie die Regenraten R_5 und R_{50} . Die Korrekturterme für die jeweiligen Schallpegel werden über die Funktionen *getL5RainRateCorrection(..)* sowie *getL50RainRateCorrection(..)* gemäss den Formeln (4.18) und (4.19) bestimmt. Schlussendlich werden die Geräuschkoeffizienten für den L_5 - sowie L_{50} -Schallpegel in unabhängige *Array-List*-Instanzen gespeichert. Die Berechnung des gewünschten Schallpegels erfolgt dann über Gleichung (4.22) nachdem der Benutzer die auszuwertenden Messpunkte spezifiziert hat.

Funktionen der Klasse *AudibleNoise*

Nach der Erstellung einer Instanz der Klasse *AudibleNoise* muss zunächst die Funktion *calculateAudibleNoise(..)* aufgerufen werden, um die Geräuschkoeffizienten zu berechnen. Die Klasse *AudibleNoise* stellt folgende Funktionen zur Verfügung:

- **void calculateAudibleNoise(double rainRateL5, double rainRateL50, double percentageOfRain)**
Berechnet die Geräuschkoeffizienten für den L_5 - sowie L_{50} -Schallpegel bei den spezifizierten Regenraten R_5 und R_{50} sowie der prozentualen Regendauer.
- **double getSoundLevelL5AtLocation(Matrix locationVector)**
Gibt den A-gewichteten L_5 -Schallpegel im übergebenen Messpunkt zurück unter Anwendung von Gleichung (4.22).
- **double getSoundLevelL50AtLocation(Matrix locationVector)**
Gibt den A-gewichteten L_{50} -Schallpegel im übergebenen Messpunkt zurück unter Anwendung von Gleichung (4.22).
- **double getSoundLevelLeqDuringRainAtLocation(Matrix locationVector)**
Gibt den nach Gleichung (4.14) berechneten Äquivalenzschallpegel $L_{eq,r}$ während Regen am übergebenen Ortspunkt zurück.
- **double getSoundLevelLeqDuringPeriodAtLocation(Matrix locationVector)**
Gibt den nach Gleichung (4.17) berechneten Äquivalenzschallpegel $L_{eq,p}$ während einer Periode am übergebenen Ortspunkt zurück.

4.4 Schlussfolgerungen

Mit der Implementation der Berechnung der akustischen Geräuschemissionen gemäss den Formeln nach EPRI ist die Programmierung der Anwendung abgeschlossen. Für beliebige Geometrien von Hochspannungsfreileitungen lassen sich nun die äquivalenten Dauerschallpegel während Regen oder während einer Periode bestimmen, um die akustischen Emissionen hinsichtlich gesetzlich vorgeschriebener Grenzwerte beurteilen zu können.

Bei Berechnung von Zweierbündeln sei darauf hingewiesen, dass die Formeln für eine horizontale Konfiguration implementiert sind. Eine vertikale Anordnung der Teilleiter erzeugt gemäss EPRI einen um 1.5 dB höheren Schallpegel [2].

Die Anwendung liefert für den Standard-Teilleitermodus genau die Resultate, welche mit den Gleichungen nach EPRI übereinstimmen. Die Verallgemeinerung der Formeln für den Expertenmodus kann mit den heutigen Angaben zum Formelwerk nicht physikalisch begründet werden, ist aber wohlmotiviert.

Abbildungsverzeichnis

1.1	Benutzeroberfläche von <i>HVLNoise</i>	3
2.1	Dialogfenster zur Angabe der Parameter eines elektrischen Systems in der Anwendung	6
2.2	xy-Koordinatensysteme zur Platzierung der Leiter resp. Teilleiter	7
2.3	Ansicht des Expertenmodus zur Platzierung der Teilleiter	7
2.4	Hierarchische Datenstruktur zur Speicherung der Leitungsparameter	8
3.1	Anordnung von Konturpunkten und Ersatzladungen auf Teilleiterkontur	17
3.2	Platzierung von Konturpunkten und Ersatzladungen sowie Zuordnung der Spiegelladungen eines Teilleiters.	19
3.3	Ersatzladung und ihre Spiegelladung zur Herleitung der Potentialkoeffizienten	20
3.4	Flussdiagramm zur Implementation der Berechnung der elektrischen Randfeldstärken	22
3.5	Skizze zur Bestimmung des maximalen Feldstärkefehlers in Abhängigkeit der Anzahl Konturpunkte	30
3.6	Fiktive Leitungsgeometrie zum Vergleich der Feldberechnung mit <i>Comsol</i> ; L1, L2, L3 Phasen des Drehstromsystems; B Bahnstromsystem; DC1, DC2 Gleichstromsystem	31
3.7	Vergleich der Randfeldstärke des linken Teilleiters von L1 mit <i>Comsol</i> und relativer Fehler	32
4.1	Kurvenfit zur Korrektur der Schallpegel zur Berücksichtigung der Regenrate	39
4.2	Grafische Oberfläche zur Wahl des Berechnungsmodus	41

Tabellenverzeichnis

3.1	Dem Benutzer zur Verfügung stehende Rechengenauigkeiten mit entsprechender Anzahl Konturpunkte	31
3.2	Koordinaten der Leiter sowie Teilleiter in m der Mastgeometrie zum Vergleich der Feldberechnung mit <i>Comsol</i>	32
4.1	Koeffizienten zum Fit des rationalen Polynoms (4.18) zur Schallpegelkorrektur gemäss Abbildung 4.1	40
4.2	Koeffizienten zum Fit des rationalen Polynoms (4.19) zur Schallpegelkorrektur gemäss Abbildung 4.1	40

Literaturverzeichnis

- [1] ANZIVINO, L.D. ; GELA, G. ; GUIDI, W.W. ; JOHNSON, G.B. ; LAFOREST, J.J. ; NICHOLLS, C.W. ; SCHNEIDER, H.M. ; ZAFFANELLA, L.E.: *HVDC Transmission Line Reference Book*. Electric Power Research Institute EPRI, 1993
- [2] COMBER, M.G. ; NIGBOR, R.J. ; ZAFFANELLA, L.E.: *Transmission line reference book - 345 kV and above*. Kap. 6, Electric Power Research Institute EPRI, 1987
- [3] KÜCHLER, Andreas: *Hochspannungstechnik, Grundlagen - Technologie - Anwendungen*. Springer, 2005
- [4] STRAUMANN, Ulrich: *Berechnung und Reduktion der tonalen Geräuschemission von Hochspannungsfreileitungen*, ETH Zürich, Dissertation, 2007