

# 1000 Antennensimulationen

**Dr. Andreas Krüger, DJ3EI**

[dj3ei@famsik.de](mailto:dj3ei@famsik.de)

## Was geht?

Wer ein Antennensimulationsprogramm mit Software steuert, kann innerhalb weniger Minuten 1000 und mehr ähnliche Antennen bequem ausprobieren.

Was geht damit? **Parameterstudien** bieten Erkenntnisgewinn, **Optimierungen** suchen an eigenen Vorlieben und Bedürfnisse angepasste Spezialantennen, und **indirekte Antennenmessung** kann basierend auf Impedanz-Messdaten (manchmal) Antennendaten ermitteln.

## Das ist nicht neu

Das Vorgehen ist nicht neu. Es wurden schon seit langem Antennensimulationen geduldig immer wieder laufen gelassen, um Ziele zu erreichen.

Zum Beispiel ist Optimierung üblich: Gängige Programmpakete bieten seit Jahrzehnten die Möglichkeit, eine Antenne zum Beispiel auf niedrigstes SWR oder höchsten Gewinn zu trimmen.

Die Sache wird aber besonders pfiffig, wenn eigener Programmcode die Kontrolle übernimmt. So können kompliziertere, spezielle Ziele verfolgt werden, die Autoren der gängigen Programmpakete nicht vorgesehen haben.

## Optimierungs-Beispiele

Dieser Artikel gibt anhand einiger weniger Beispiele einen ersten Einblick, was mit diesem Ansatz, mit eigenem Programmcode, erreichbar ist.

Natürlich bin ich nicht als Erste(r) im Amateurfunk auf diese Idee gekommen. Für Programmierer ist sie einfach zu naheliegend. Daher gibt es viele Vorbilder. Nur zwei herausgegriffen:

- Jan, DL9JBE hat vor etwa sieben Jahren eine Antenne entwickelt, die zirkular polarisierte Wellen aussendet und empfängt. Die Antenne basiert nicht, wie sonst üblich, auf zwei linear polarisierten Antennen, die mit 90° Phasenunterschied gespeist werden. Statt dessen sieht sie zunächst ähnlich aus wie eine normale Yagi-Uda-Antenne, allerdings sind ihre Elemente schraubenartig verdreht mit dem Boom als Drehachse. Details hat [1].
- Bernd, DO3YQ, hat Ostern 2025 beim CCC-Szenetreff „Easterhegg“ eine Dipolantenne mit stark verbogenen Ästen vorgestellt, die mit guter Näherung in alle Richtungen in gleicher Feldstärke sendet, also einen Isotropenstrahler approximiert<sup>1</sup>, siehe [2].

Doch zunächst eine Erklärung, wie eigener Programmcode die Kontrolle übernehmen kann:

## Aufbau üblicher Antennensimulationsprogramme

Die im Amateurfunk üblicherweise benutzten Programmpakete für Antennensimulation bestehen aus zwei Teilen: Eine GUI („Graphical User Interface“, also Bedienoberfläche) und dem eigentlichen Simulationskern. Die GUI nimmt vom Benutzer entgegen, welche Antenne simuliert werden

---

<sup>1</sup> Das setzt voraus, dass bei einem Isotropenstrahler elliptische Polarisierung erlaubt ist. Ist nur lineare erlaubt, gibt es keinen Isotropenstrahler.

soll: Das umfasst Geometrie und Lage, Eigenschaften und Material der Leiter, Bodenparameter und natürlich die Information, bei welcher Frequenz oder welchen Frequenzen die Antenne überhaupt simuliert werden soll.

Diese Daten werden dem Simulationskern übergeben. Der führt die eigentlichen Berechnungen durch und liefert die Ergebnisse zurück an die Bedienoberfläche zur Anzeige. Zu diesen Ergebnissen gehören beispielsweise die Antennenimpedanz, die Antennen- und Erdverluste sowie Informationen zum Strahlungsfeld.

Bild 1 zeigt das soeben textuell beschriebene als schematische Grafik.

Die in diesem Artikel vorgestellte Vorgehensweise basiert darauf, dass der Teil „Bedienoberfläche“ ersetzt wird durch ein (selbst verfasstes) „Steuerprogramm“. Es kann den Simulationskern mit vielen Antennenvarianten laufen lassen, etliche Male in der Sekunde, und die jeweiligen Ergebnisse entgegennehmen und verarbeiten. Bild 2 zeigt das wieder als Grafik; gegenüber Bild 1 wurde nur die „Bedienoberfläche“ ausgetauscht gegen das „Steuerprogramm“.

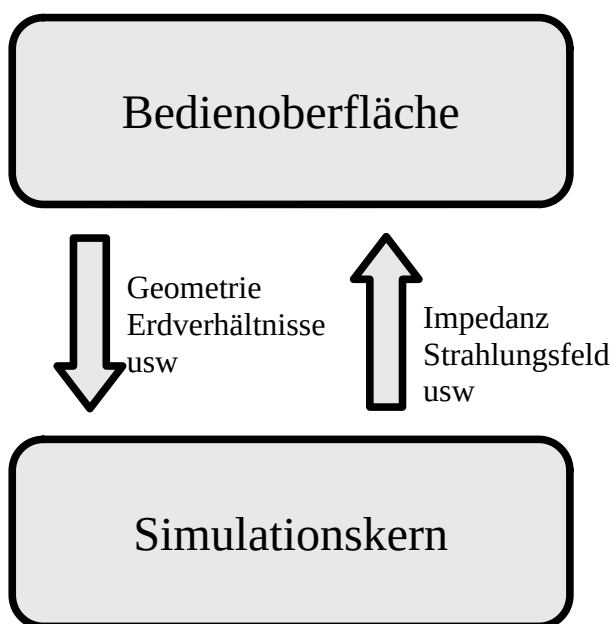


Bild 1: Aufbau üblicher Antennensimulationspakete

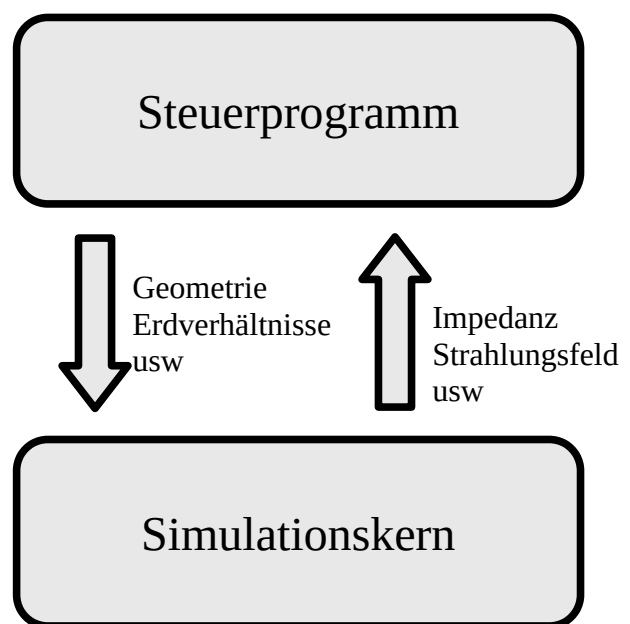


Bild 2: Aufbau mit eigenem Steuerprogramm

## Simulationskerne

Der einschlägige englischsprachige Wikipediaartikel [3] verschafft einen Überblick über übliche Simulationskerne. Der im Jahre 1981 entwickelte NEC2-Kern war von Anfang an ein Open Source Programm (FLOSS) und konnte daher im Laufe der Zeit immer wieder auf neue Rechner und auch in andere Programmiersprachen übertragen werden.

Derzeit wird an einem Simulationskern „Arcanum“ [4] gearbeitet, ebenfalls FLOSS, der von Grund auf neu programmiert wird.

Ich persönlich nutze das Programm nec2++ aus dem gleichnamigen FLOSS-Projekt [5]. Es handelt sich dabei um eine Portierung des NEC2-Simulationskerns in die Programmiersprache C++.

## Mein Projekt antenna-simulation-driver

Für die Kommunikation zwischen Steuerprogramm und dem nec2++ Simulationskern (die Pfeile in Bild 2) müssen Daten hin und her fließen. Wenn das Steuerprogramm in der gängigen Programmiersprache Python geschrieben ist, kann an dieser Stelle eine FLOSS-Software antenna-simulation-driver helfen, die ich über PyPI zur Verfügung stelle, siehe [7]. Diese Software startet den Simulationskern, übergibt ihm vom eigentlichen Steuerprogramm erzeugte Daten, nimmt die Ergebnisse

des Simulationskerns entgegen und stellt sie dem Steuerprogramm in sinnvollen Datenstrukturen organisiert zur Verfügung.

Dabei nutzt meine Software eine Schnittstelle, die Texte in den Simulationskern hineinschauft und Texte aus ihm wieder entgegen nimmt und einliest. Das sind dieselben Texte, die der Simulationskern auch als Eingabedatei von der Festplatte lesen könnte bzw. als Ausgabedatei dorthin schreiben. Viel Fleißarbeit, die als Text vorhandenen Simulationsergebnisse sinnvoll einzulesen, wird von meiner Software erledigt.<sup>2</sup>

## Textbasierte Entgegennahme der Simulationsergebnisse: Vor- und Nachteile

Dieses textbasierte Vorgehen hat Vor- und Nachteile. Zu den Vorteilen zählt, dass es relativ leicht ans Laufen zu kriegen war. Wenn etwas nicht funktioniert, lässt sich bequem als Text auffangen, was der Simulationskern als Eingabe bekommen und als Ausgabe geantwortet hat: Das erleichtert eventuell notwendige Fehlersuche. Für jede Simulation wird das Programm nec2++, das den Simulationskern zur Verfügung stellt, neu gestartet: Damit ist das Risiko minimal, dass Überreste früherer Berechnungen spätere Berechnungen ungewollt beeinflussen. Da sich das Eingabe- und Ausgabeformat der verschiedenen NEC2-Varianten seit Jahrzehnten nicht wesentlich geändert hat, kann ich auf vorhandene allgemeine Dokumentation zurückgreifen (leider ist spezifische Dokumentation der Internas von nec2++ eher dürftig). Wenn Arcanum demnächst zur Verfügung steht, sollte es sehr leicht sein, den nec2++-Simulationskern dagegen auszutauschen oder sogar beide Simulationskerne für Vergleichszwecke gegeneinander antreten zu lassen.

Aber es gibt auch Nachteile: Weil das Programm nec2++ für jede einzelne Antenne komplett neu gestartet wird, ist die Sache verhältnismäßig langsam.<sup>3</sup> Der Simulationskern rundet jede einzelne Zahl, ehe er sie als Text ausgibt, dadurch sinkt die Genauigkeit der Zahlen, die im Steuerprogramm zur Verfügung stehen. Diese Nachteile wären vermieden, würden Simulationskern und Steuerung zu einem Programm verschmelzen. Für die Pfeile aus Bild 2 könnte dann interne, direktere Datenübertragung benutzt werden. Tatsächlich unterstützt nec2++ so ein Vorgehen, indem es eine interne Python-Schnittstelle anbietet. Ich hatte versucht, die für mich ans Laufen zu bekommen. Ich hatte damit aber Probleme, fand keine gute Dokumentation für die Schnittstelle, mochte keine Zeit in Detektivarbeit im Sourcecode von nec2++ investieren und bin dann auf das beschriebenen textbasierten Vorgehen ausgewichen.

## Anwendungen

Nach dieser Einführung stellt der Rest dieses Artikels die schon erwähnten drei Anwendungsmöglichkeiten anhand von Beispielen vor:

- Was passiert, wenn Dipolantennen kurz und immer kürzer werden? Eine **Parameterstudie** gibt Antworten.
- Als Beispiel für eine **Antennenoptimierung** wird die bekannte ZS6BKW-Multibandantenne neu gesucht (und gefunden!).
- **Indirekte Antennenmessung** kann (manchmal) helfen, wenn von einer Antenne nur bekannt ist, wie sie grundsätzlich aussieht, aber einige Abmessungsdaten fehlen. Liegen wenigstens einige SWR- oder Impedanzmesswerte vor, kann versucht werden, daraus Abmessungsdaten indirekt zu ermitteln.

---

2 Allerdings ist die Umsetzung nicht vollständig. Es gibt Ergebnisse, die vom Simulationskern angefordert werden können, die meine Software aber derzeit nicht einlesen kann.

3 Was heißt „verhältnismäßig“? Mein handelsüblicher, gebraucht gekaufter, etwa 5-6 Jahre alter Laptop mit Quad-core-Prozessor schafft es, etwa 8-14 einfache Dipolantennen pro Sekunde zu simulieren, wenn Informationen über das Strahlungsfeld nicht besonders detailliert verlangt werden.

## Parameterstudie: Kurze Dipole

In kurzer Zeit relativ viele Antennen simulieren zu können kann helfen, Antennen zu verstehen.

Welche will ich verstehen? Kurze Dipole.

Denn als Bewohner einer Großstadt-Mietwohnung habe ich nur beschränkte Möglichkeiten, Antennen aufzuhängen. Es läuft bisher meist auf Dipole hinaus, wobei ich schon für mein Brot-und-Butter-Band 40 m den Platz für einen Halbwellendipol nicht habe. Da frage ich mich: Was passiert eigentlich, wenn ich einen kürzeren Dipol aufhänge und mit Tuner ansteuere? Tut das weh?

Aus der Antennentheorie bekannt ist, wie sich mit der Verkürzung die komplexe Impedanz der Antenne ändert. Je kürzer die Antenne wird, um so kleiner wird der Realteil, gleichzeitig steigt der Imaginärteil. Wer sich mit komplexen Impedanzen nicht so gut auskennt: Das bedeutet, dass die Antenne sich verhält wie eine Serienschaltung aus einem Widerstand mit immer weniger Ohm und einem („virtuellen“) Kondensator mit immer weniger Picofarad.

An den Widerstand muss die Sendeleistung abgegeben werden. Da der Widerstand bei kürzerem Dipol immer kleiner wird, werden dazu bei gleicher Sendeleistung immer höhere Ströme benötigt.

Gleichzeitig müssen bei immer kürzerem Dipol diese Ströme durch den immer kleiner werdenden Kondensator geprügelt werden. Dazu werden immer höhere Spannungen gebraucht. Bild 3 und Bild 4 zeigen die Verhältnisse, wenn die Antenne (gerade einmal!) 5 W aufnehmen soll.<sup>4</sup>

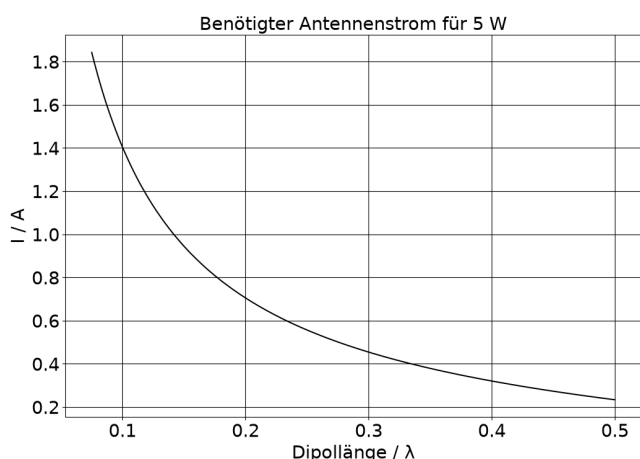


Bild 3: Benötigter Antennenstrom für 5 W

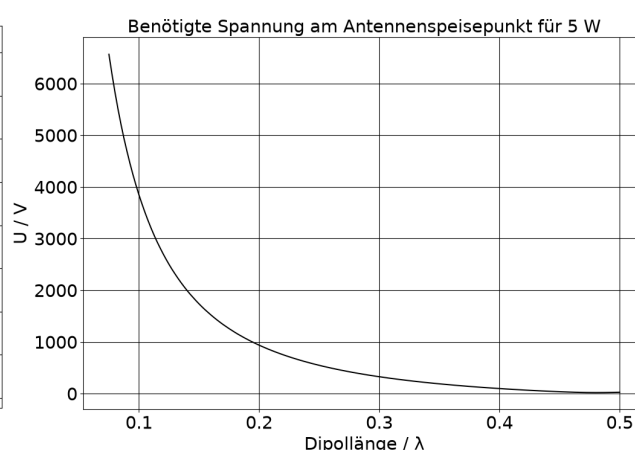


Bild 4: Benötigte Spannung für 5 W

Wenn die Dipollänge kürzer und kürzer wird, wächst der benötigte Antennenstrom noch relativ moderat an. Irgendwann explodieren aber die Spannungen, die nötig sind, damit die Antenne Leistung aufnimmt. Bei einer Gesamt-Dipollänge von etwa  $0,2 \lambda$  ist bereits eine Spannung von 1 kV am Speisepunkt nötig, um nur 5 W Leistung an die Antenne los zu werden.

Glücklicherweise lassen sich die hohen Spannungen relativ leicht erzeugen, indem der („virtuelle“) Kondensator mit einer Spule zu einem Serienschwingkreis ergänzt wird.

Ich persönlich mache mir noch keine großen Sorgen, wenn ich mit einem Antennentuner einen Dipol anpasse, dessen Gesamtlänge größer als  $\lambda/4$  ist. Salopp formuliert: Halb so lang wie ein Halbwellendipol geht noch. Mit dieser Länge bleibt das Dipol noch im Bereich vor der „Spannungsexplosion“.

## Antennenverluste

Höherer Strom führt zu höheren Leitungsverlusten im Antennenleiter. Das wird immerhin etwas dadurch kompensiert, dass kürzere Antennen kürzere Antennenleiter haben.

---

4 Den beiden Bildern liegen Impedanzwerte aus 170 Antennensimulationen zugrunde.

Unangenehm sind die Erdverluste. Die hohen Spannungen führen zu starken elektrischen Feldern, die in der Erde zu höheren Verlusten führen.<sup>5</sup> Bild 5 vergleicht die Verluste bei schlechter Innenstadterde mit denen bei durchschnittlicher Erde. Hier wurden Dipole aus Draht mit 1 mm Durchmesser 10 m über Grund bei 7,1 MHz simuliert.<sup>6</sup>

## Tunerverluste

Wer kurze Dipole an seinen TX anschließen will, benötigt in der Regel einen (wie auch immer gear- teten) Antennentuner. Der hat die Aufgabe, die Antennenimpedanz auf 50  $\Omega$  zu transformieren. Denn 50  $\Omega$  ist die Lastimpedanz, für die unsere Sender normalerweise entworfen wurden.

Beispielhaft betrachte ich im Folgenden den klassischen L-Tuner, siehe Bild 6. Er bestehend aus einer Spule in Serie mit der Antenne und einem Kondensator parallel zur dieser Serienschaltung. Er heißt so, weil sich die beiden Bauteile zusammen als ein um 90° im Uhrzeigersinn gedrehtes „L“ vorstellen lassen. Bei zweckmäßiger Dimensionierung ergibt sich parallel zum Kondensator die gewünschte Impedanz von 50  $\Omega$ . Bei den Impedanzen, die bei kurzen Dipolen auftreten, lassen sich passende Werte für L und C immer finden.<sup>7</sup> Die Spule L übernimmt dabei gleichzeitig die Aufgabe, den („virtuellen“) Kondensator der Antenne zu einen Serienschwingkreis zu ergänzen und ihn einerseits, von außen betrachtet, weg zu kompensieren, andererseits erzeugt der Serienschwingkreis schwingkreisintern die nötige Spannung.

Ein Maß für die Verluste einer gegebene Spule einer bestimmten Induktivität bei bekannter Belastung bietet die Spulengüte Q.<sup>8</sup> Ich gebe zu, mir noch nie die Mühe gemacht zu haben, die Güte einer Spule tatsächlich zu messen. Der Literatur entnehme ich den Eindruck, dass für Kurzwellenzwecke eine Spule der Güte Q = 150 relativ einfach herzustellen ist, mit Sorgfalt auch 200-250 erreichbar sind und wer weiß, was er tut und nichts dem Zufall überlässt, kann vielleicht sogar eine große, durchoptimierte Rekordspule herstellen mit einer Güte von 1000<sup>9</sup>.

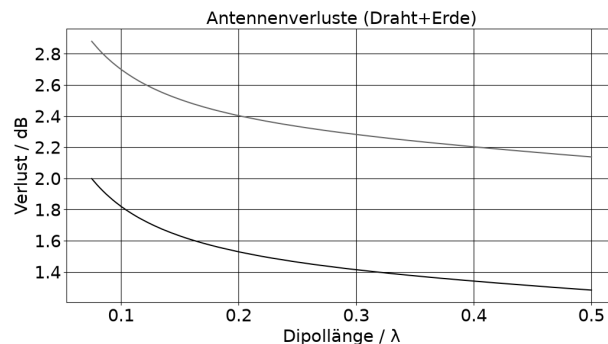


Bild 5: Antennenverluste bei Innenstadterde (oben) und durchschnittlicher Erde (unten), siehe Text.

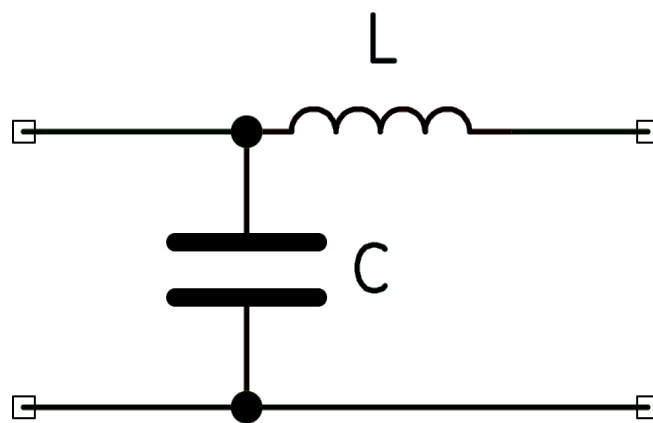


Bild 6: L-Tuner. TRX links, kurzes Dipol rechts angeschlossen

5 Und übrigens führen die enormen Spannungen und dadurch bedingten starken elektrischen Nahfelder auch zu Verlusten in irgendwelchen Gegenständen, die sich in diesen Feldern befinden. Dabei handelt es sich teilweise um ohmsche Verluste, wenn die elektrischen Felder Ströme induzieren, und teilweise um Dielektrikumverluste. Wegen der starken Nahfelder sind kurze Antennen auch grundsätzlich problematisch, was EMV, EMU, Personenschutzabstände und ähnliches betrifft. Das betrifft alle kleinen Antennen, auch zum Beispiel Magloops (bei denen das Problem aber magnetische Nahfelder sind, nicht elektrische).

6 Hier wurden die Ergebnisse weiterer 170 Antennensimulationen ausgewertet.

7 Eine Impedanz  $x - jy$  ( $x$  und  $y$  beide positiv) lässt sich mit dem beschriebenen L-Tuner auf 50  $\Omega$  transformieren, wenn  $x$  kleiner als 50  $\Omega$  ist. Solche transformierbaren Impedanzen treten bei allen halbwegs kurzen Dipolen auf.

8 Einzelheiten zur Spulengüte Q finden sich in den Folien [6] eines Vortrags, den ich 2022 auf der HAM RADIO gehalten habe. Es sind ziemlich viele Folien, aber sie sind so konzipiert, dass sie für (geduldiges) Selbststudium geeignet sind. Für den Herbst 2026 ist übrigens eine Serie von Online-Veranstaltungen geplant, die sich ebenfalls mit diesen Themen beschäftigt; darin enthalten ein kleiner online Python-Programmierungskurs „Filterkurven selbst berechnen“.

9 Ein Job, auf den ich mich nicht bewerbe. – Bei höheren Frequenzen als Kurzwelle sind hohe Spulengüten einfacher erreichbar.

Die Güte des Kondensators C ist normalerweise deutlich höher als die der Spule L. Daher sind für die Tunerverluste die Verluste in der Spule wesentlich, die im Kondensator können vernachlässigt werden.<sup>10</sup>

Mit diesem Hintergrundwissen können wir nun wieder den Computer für uns arbeiten lassen.

Vielleicht empfinden wir einen Verlust von 1 dB als noch harmlos. Er entspricht einem Wirkungsgrad von 79 %. Ich jedenfalls zahle einen Preis von 79 % Wirkungsgrad im Tuner gerne, wenn ich dadurch trotz meiner beengten Platzverhältnisse auf einem tiefen Band überhaupt arbeiten kann.

Betrachten wir nun beispielhaft Dipolantennen verschiedener Länge mit folgenden Details: Simuliert werden sie bei 7,1 MHz und sie bestehen aus Kupferdraht von 1 mm Durchmesser in 10 m Höhe über durchschnittlichem Boden. Schließlich werden sie mit einem L-Tuner angepasst. Welche Spulengüte wird gebraucht, um die Tunerverluste auf 1 dB zu beschränken?

Die Antwort gibt Bild 6.

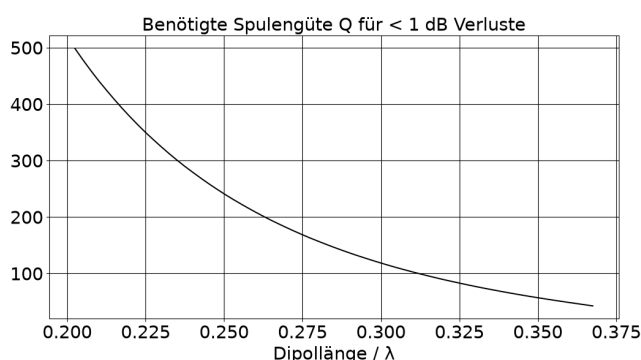


Bild 6: Spulengüte für < 1 dB Verlust

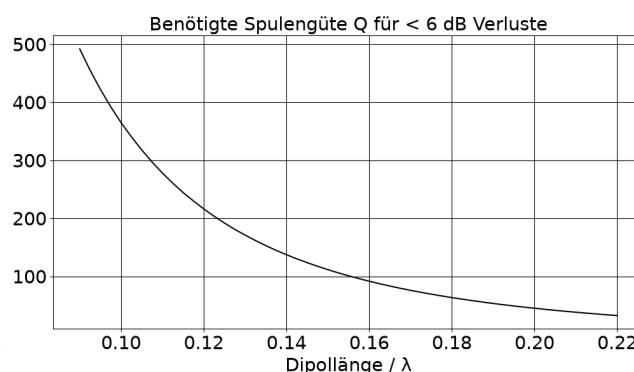


Bild 7: Spulengüte für < 6 dB Verlust

Übrigens finde ich in diesem Bild die oben schon erwähnte Faustregel bestätigt, dass es ungefähr bei einer Dipol-Gesamtlänge von  $\lambda/4$  (also ab einem halbierten Halbwellendipol) anfängt, schwierig zu werden.

79 %? Andere zahlen noch ganz andere Preise! Wer auf einem Band lieber mit 25 % Wirkungsgrad QRV wird als gar nicht, also mit 6 dB Tunerverlusten zu leben bereit ist, kann mit realistischen Spulen sehr viel kürzere Dipole tunen. Bild 7 gibt Auskunft.<sup>11</sup> Auch mit verzweifelt kurzen Dipolen der Gesamtlänge  $\lambda/8$  oder gar  $\lambda/10$  lassen sich QSOs führen – viel viel besser als ganz ohne Antenne!<sup>12</sup>

## Mehr Parameterstudien-Material

Das hier gegebene Material stammt aus einer größeren Studie verschiedener Dipolantennen für beengte Platzverhältnisse. Sie heißt „Die hwg-Antenne“, wobei „hwg“ die Abkürzung ist für „hängen, was geht“. Dort wird auch die „inverted U“ oder „Hängeohrenantenne“ vorgestellt, die einen anderen Ausweg weist aus Situationen, wo zwar horizontal wenig Platz ist, aber vielleicht vertikal Drähte ein Stockwerk oder mehr nach unten baumeln dürfen. Diese Studie ist Bestandteil des Schaufensters [8] meines Programmpaketes.

## Antennenoptimierungsziel: Die ZS6BKW Multiband-Antenne

Brian Austin, G0GSF, vorher ZS6BKW, hat eine Multibandantenne erfunden, die nur aus einem mittengespeisten Dipol der Spannweite  $L_1$  und einer Zweidraht-Speiseleitung der Länge  $L_2$  besteht.

<sup>10</sup> Die hier gegebene Besprechung von Tunerverlusten ist korrekt und bietet einen Einstieg ins Thema, ist aber nicht umfassend. Variabel einstellbare Tuner haben je nach Schaltung höhere Verluste. Weitere Verluste können entstehen, wenn der Tuner über eine längere Koaxleitung mit der Antenne verbunden ist.

<sup>11</sup> Für Bild 6 und Bild 7 wurden die ursprünglichen 170 Simulationsergebnisse neu ausgewertet.

<sup>12</sup> Übrigens sind sehr kurzen Vertikalantennen mit ähnlichen Verlusten verbunden. Viele Ops haben mit solchen Antennen trotzdem QSOs und ihren Spaß. Manche wissen, wie viel sie heizen und wie wenig sie senden, andere nicht.

Durch geschickte Wahl von  $L_1$  und  $L_2$  ergibt sich ein SWR unter 2 auf fünf Amateurfunkbändern (nämlich von 40 m bis 10 m ohne 30 m und 15 m). Eher zufällig ist diese Antenne sogar auch im 6 m-Band ohne Tuner brauchbar.

Brians ursprüngliches Anliegen waren zuverlässige Kurzwellenverbindungen im Bereich um 1000 km, vermutlich für die südafrikanische Bergbauindustrie (Goldminen). In den 1980er Jahren hat er zu diesem Zweck einfache, robuste Antennen entwickelt, die ohne Weiteres auf mehreren Bändern betreibbar sind und so bequem Bandwechsel je nach MUF-Bedingungen ermöglichen. Die Veröffentlichung [9] seiner Ergebnisse listet abstrakt Quotienten von Frequenzen auf, die von solchen Antennen gleichzeitig mit  $\text{SWR} < 2$  bedient werden können. Die Amateurfunkantenne, die wir heute als ZS6BKW kennen, ist ein Abfallprodukt dieser Forschung.

Brian hat damals teilweise manuell gearbeitet, unter anderem mit dem Smith-Chart, aber parallel auch computergestützte automatische Optimierung benutzt. Allerdings wurden damals, Anfang der 1980er, die Antennensimulationskerne erst noch entwickelt. Für seine computergestützten Arbeiten hat Brian daher den Simulationskern aus Bild 2 durch eine Art Datenbank ersetzt, aus der fertig berechnete Dipolimpedanzen entnommen wurden.<sup>13</sup> Tabellen von Dipolimpedanzen<sup>14</sup> waren damals in entsprechenden Veröffentlichungen zugänglich.

|                                        | <b>B pub</b> | <b>B sim</b> | <b>1 mm</b> | <b>Ø opt</b> | <b>2 mm</b> |
|----------------------------------------|--------------|--------------|-------------|--------------|-------------|
| <b><math>L_1</math> / m</b>            | 28,5         | 28,5         | 28,787      | 28,516       | 28,568      |
| <b><math>L_2</math> / m</b>            | 13,3         | 13,3         | 13,329      | 13,445       | 13,438      |
| <b>40 m: <math>f_b</math> / MHz</b>    | 7,10         | 7,115        | 7,073       | 7,113        | 7,108       |
| <b>40 m: <math>\text{SWR}_b</math></b> | 1,1          | 1,01         | 1,00        | 1,12         | 1,12        |
| <b>40 m: B / kHz</b>                   | 360          | 347          | 346         | 352          | 352         |
| <b>20 m: <math>f_b</math> / MHz</b>    | 14,20        | 14,342       | 14,244      | 14,209       | 14,197      |
| <b>20 m: <math>\text{SWR}_b</math></b> | 1,1          | 1,08         | 1,05        | 1,10         | 1,09        |
| <b>20 m: B / kHz</b>                   | 270          | 247          | 246         | 255          | 255         |
| <b>17 m: <math>f_b</math> / MHz</b>    | 18,10        | 18,125       | 18,028      | 18,049       | 18,040      |
| <b>17 m: <math>\text{SWR}_b</math></b> | 1,3          | 1,08         | 1,06        | 1,25         | 1,24        |
| <b>17 m: B / kHz</b>                   | 380          | 393          | 393         | 385          | 387         |
| <b>12 m: <math>f_b</math> / MHz</b>    | 24,92        | 25,099       | 24,917      | 24,928       | 24,905      |
| <b>12 m: <math>\text{SWR}_b</math></b> | 1,4          | 1,49         | 1,55        | 1,41         | 1,43        |
| <b>12 m: B / kHz</b>                   | 260          | 231          | 214         | 255          | 251         |
| <b>10 m: <math>f_b</math> / MHz</b>    | 28,97        | 29,127       | 28,982      | 28,957       | 28,946      |
| <b>10 m: <math>\text{SWR}_b</math></b> | 1,4          | 1,16         | 1,12        | 1,37         | 1,36        |
| <b>10 m: B / kHz</b>                   | 400          | 436          | 439         | 412          | 416         |
| <b>6 m: <math>f_b</math> / MHz</b>     | 51           | 51,258       | 51,028      | 50,884       | 50,871      |
| <b>6 m: <math>\text{SWR}_b</math></b>  | 1,5          | 1,18         | 1,15        | 1,44         | 1,42        |
| <b>6 m: B / kHz</b>                    |              | 481          | 481         | 445          | 450         |

Die nebenstehende Tabelle gibt einen Überblick über verschiedene Varianten der ZS6BKW-Antenne.  $L_1$  ist jeweils die Dipolspannweite,  $L_2$  die elektrische Länge der benutzten Zweidraht-Speiseleitung ohne Berücksichtigung des Verkürzungsfaktors VF. In Brians Veröffentlichung [10] wird eine Speiseleitung mit Impedanz (Wellenwiderstand) von  $400\ \Omega$  benutzt. Dieser Wert wird hier einheitlich für alle Varianten übernommen. Die Höhe ist immer 10 m über Grund. Für die Bodenverhältnisse nennt Brian „typical urban ground“; ich füttere NEC2 mit der Dielektrizitätskonstante 3 und der Leitfähigkeit  $0,001\text{S/m}$ . Für jedes der fünf Bänder ist die Frequenz  $f_b$  des besten SWRs angegeben, das bei  $f_b$  erreichte beste  $\text{SWR}_b$  und die Bandbreite B des Intervalls mit  $f_b$  in der Mitte, innerhalb dessen das SWR unter 2 bleibt.<sup>15</sup>

<sup>13</sup> Für Fachleute: Die Veröffentlichung [9] spricht von „Lookup Table“.

<sup>14</sup> Im Freiraum!

<sup>15</sup> Streng genommen können hierbei die Frequenz des besten SWRs und die Mittenfrequenz des Bereichs mit  $\text{SWR} < 2$  geringfügig voneinander abweichen. Das wird ignoriert.

Die von Brian in [10] publizierten Werte finden sich in der ersten Wertespalte „B pub“<sup>16</sup>. Wenn ich Brians Antenne mit meinem heutigen NEC2-Setup simuliere, kommen die Werte in der Spalte „B sim“ heraus. Die Frequenzen liegen höher als gewünscht, augenscheinlich ist die Antenne etwas zu kurz.

## ZS6BKW durch Optimierung neu finden!

Also optimieren!

Gebraucht wird für eine Antennenoptimierung ein Steuerprogramm, das verschiedene Werte mehr oder weniger systematisch ausprobiert und so die goldene Nadel aus dem Heuhaufen herausucht. So etwas kann selbst programmieren, wer Lust darauf hat. Ich habe in die Schatztruhe vorhandener FLOSS gegriffen: Die bekannte Python-Software SciPy [11] bietet unter der Überschrift „global optimization“<sup>17</sup> mehrere für den Zweck potentiell nutzbare Algorithmen an, ich nutze „differential evolution“<sup>18</sup>. Dieser Algorithmus scheint für unsere Zwecke ganz brauchbar zu funktionieren.<sup>19</sup> Möglicherweise funktionieren andere noch besser, das habe ich nicht untersucht.

Ich habe dem Algorithmus vage Vorgaben gemacht: Die Dipolspannweite soll irgendwo zwischen 6 und 100 m liegen, die elektrische Feedlinelänge zwischen 1 und 40 m. Die Aufgabe ist, die Summe der  $(s_i-1)^2$  zu minimieren, wobei  $s_i$  für den SWR-Wert bei der  $i$ -ten der fünf  $f_b$  Kurzwellenfrequenzen von Brian aus der Spalte B pub steht.

Mit diesen vagen Vorgaben kam eine Antenne mit Dipolspannweite 28,787 m und Feedlinelänge 13,329 m heraus. Offensichtlich hat der Algorithmus die ZS6BKW-Multibandantenne wiederentdeckt! Die im Einzelnen erreichten Daten finden sich in der Spalte „1 mm“.<sup>20</sup>

## Indirekte Antennenmessung: Drahtdurchmesser

Warum nenne ich diese Spalte „1 mm“?

Ich nutze als Draht für meine Antennenexperimente meistens auseinandergezogene ursprünglich zweiadrige billige Lautsprecherlitze aus dem Baumarkt. Ihr Leiterquerschnitt von 0,75 mm<sup>2</sup> entspricht rechnerisch einem Leiterdurchmesser von 0,98 mm, also fast genau 1 mm. Brian macht in [10] keine Angaben zum verwendeten Dipolleiter. Also habe ich meine übliche 0,98 mm-Litze zugrunde gelegt.

Mir kam dann der Verdacht, dass Brians Daten auf einem anderen Drahtdurchmesser basieren. Ich habe daraufhin einen neuen Optimierungsalgorithmus gestartet. Der sollte basierend auf Brians Maßen den Drahtdurchmesser suchen, bei dem die von Brian publizierten SWR-Werte möglichst genau reproduziert werden. (Die Aufgabe war diesmal also nicht, eine Multiband-Antenne zu finden, die die SWR-Werte minimiert, sondern die Aufgabe war, eine Antenne zu suchen, die die fünf Kurzwellen-SWR-Werte aus der Spalte „B pub“ möglichst genau reproduziert.) Das hat zu keinem brauchbaren Ergebnis geführt.

---

16 Dort findet sich keine Bandbreitenangabe für das 6 m-Band.

17 Siehe <https://docs.scipy.org/doc/scipy/reference/optimize.html#global-optimization>

18 ... sowie bei der SWR < 2 Bandbreitenbestimmung den Algorithmus toms748.

19 Etwas unangenehm fällt auf, dass „differential evolution“ nicht immer den besten im Laufe der Suche ausprobierten Wert als Ergebnis liefert, sondern oft einen (geringfügig) schlechteren. Das lässt sich durch eigenen Code leicht verbessern, der sich schlicht den besten bisher gefundenen Wert merkt.

20 Für dieses Ergebnis simulierte der Algorithmus 258 Dipole unterschiedlicher Längen, jeden für 5 Frequenzen, insgesamt wurden dafür also 1290 Antennensimulationen durchgeführt. Es wurden jeweils 5 Simulationskerne gleichzeitig gestartet, um die Multicore-CPU meines Laptops gut auszunutzen. (NEC2 bietet an sich die Möglichkeit, Daten einer Antenne für mehreren Frequenzen *in einem Lauf* zu ermitteln, die dabei entstehenden Daten kann die derzeitige Version meines Programms antenna-simulation-driver aber nicht einlesen.) Für jeden einzelnen der 258 Dipole suchte dann eine „innerer Optimierung“ die optimale Feedlinelänge, die die Summe der fünf  $(s_i-1)^2$  für diesen Dipol minimiert. Auch diese inneren Optimierung nutzt wieder „differential evolution“. Die ganze Optimierung dauerte insgesamt etwa anderthalb Minuten.

Dann habe ich dem Algorithmus bei gleicher Aufgabenstellung erlaubt, außer dem Drahtdurchmesser auch die Dipolspannweite und die Feedlinelänge zu manipulieren.

Der Algorithmus finden dann einen Drahtdurchmesser von 1,97 mm, eine Dipolspannweite von 28,516 m und einer Feedlinelänge von 13,445 m. Die einzelnen Daten finden sich in der Spalte „Ø opt“. Sowohl die Dipolspannweite als auch die meisten  $f_b$ -Frequenzen sind spektakulär gut getroffen.

Es war hier also möglich, aus einigen SWR-Werten Antennenmaße indirekt zu bestimmen. Noch besser ginge das natürlich mit komplexen Impedanzwerten.<sup>21</sup>

Der Vollständigkeit halber habe ich dem Algorithmus als nächstes die Aufgabe gegeben, eine möglichst gute Multibandantenne für 1,97 mm Drahtdurchmesser zu finden. Nun sollen also nicht mehr die von Brian publizierten SWR-Werte getroffen werden, sondern möglichst niedrige SWR-Werte waren wieder das Ziel, wie auch schon für die „1 mm“-Antenne. Das Ergebnis findet sich in der Spalte „2 mm“.

## Geht da noch mehr?

Die ZS6BKW-Antenne kratzt nur an der Oberfläche dessen, was an Optimierungen möglich ist. Sie ist auf zwei Arten beschränkt:

Zum einen hat der Optimierungsalgorithmus nur zwei Variablen zur Verfügung: Dipolspannweite und Feedline-Länge. Es gibt also nur zwei Stellschrauben, an denen gedreht werden kann. Außer-mittige Speisung könnte zum Beispiel interessante optimierte Stromsummenantennen ergeben.

Zum Zweiten ist die ZS6BKW eine relativ universelle Antenne. Auf meinem persönlichen Computer kann ich aber auch eine Antenne entwerfen, die wenig universell auf meine persönlichen Verhältnisse abgestimmt ist.

## Abschluss: Etwas zum Schmunzeln

Ich hätte zum Abschluss dieses Artikels gerne so eine persönliche Antenne vorgestellt, die ich für mich optimiert, entworfen und aufgebaut habe. Doch beim ersten Versuch bin ich in eine Sackgasse gelaufen und hatte dann keine Zeit mehr für einen zweiten.

Als ersten Versuch wollte ich einen außermittig gespeisten Dipol optimieren, mit einem Hängeohr nach unten am längeren Dipolast und mit 300  $\Omega$  Bandleitung vom Speisepunkt am Fenster zu meinem Stationstisch. Als zusätzliche Stellschraube hatte ich dem Algorithmus eine zum Speiseleitungsende auf dem Stationstisch parallele Induktivität angeboten, deren Wert er bestimmen sollte. Das war ein spontanes Experiment, ich hatte mir eigentlich nicht viel davon erhofft.

Aber Holla! Das Ergebnis war ziemlich sofort eine phantastische Traumantenne! Nachdem ich immer unbescheidenere Wünsche im Dialog mit der Optimierung präzisiert hatte, blieb im unglaublich breiten Bereich von 3,1 bis 11,6 MHz das SWR unter 2. Das versprach Betrieb auf gleich vier sonst schwierigen Bändern: 80 m, 60 m, 40 m, 30 m. Dazu kamen noch das ganze 20 m - Band und große Teile von 10 m! Wow!

Diese Traumantenne habe ich probenhalber aufgebaut. Leider verhielt sich das entstehende reale Gebilde komplett anders als vom Programm vorhergesagt. Stundenlang versuchte ich dieses und jenes, schlief eine Nacht drüber, experimentierte weiter, machte Messungen, versuchte, mit Programmhil-

---

21 Ein wagemutiger Versuch endete mit gemischten Ergebnissen: Es ging darum, aus 8 komplexen Impedanzwerten einer Doublet-Antenne Dipol-Spannweite, Impedanz und Länge der Zweidrahtleitung, Höhe über Grund und sogar die beiden Bodenparameter alle auf einmal zu bestimmen. Die Speiseleitungslänge und -Impedanz waren hervorragend getroffen; Höhe und Bodenparameter nicht nachprüfbar, aber plausibel; aber der Algorithmus schätzte die Dipolspannweite auf 12,34 m, während die mechanische Länge 12,78 m betrug und wegen PE-Ummantelung eigentlich eine gegenüber der mechanischen *verlängerte* elektrische Länge zu erwarten war. Während der SWR-Messung hing die Antenne im Geäst zwischen Bäumen, was die Messwerte verfälscht haben mag.

fe zu den Messungen passende Antennen zu finden: Nichts funktionierte. Die Antenne verhielt sich partout nicht wie vorhergesagt.<sup>22</sup>

Endlich habe ich, misstrauisch geworden, mein Steuerprogramm noch einmal gründlich überprüft. Und da fand ich den Fehler: Die Impedanz der abschließenden Parallelinduktivität hatte ich mit der Formel  $2\pi fL$  berechnet. Da fehlte das „j“! Ich hatte also einen parallelen *Widerstand* simuliert, dessen Widerstandswert proportional mit der Frequenz steigt. Damit lässt sich leicht ein niedriges SWR über große Bandbreiten erreichen!

## Lizenz

Der Inhalt dieses Artikels darf von jedermann unter der Lizenz CC BY-SA 4.0 genutzt werden. Siehe [14] und [15].

## Referenzen

- [1] Jan Behrens, DL9JBE: „Die ungewöhnliche Twisted-Yagi-Antenne“, zuerst erschienen im Magazin „HAMSPIRIT.de“ im August 2019, auch <https://www.hamspirit.de/11230/>
- [2] Bernd Fix, DO3YQ: „Einfluss stark gebogener Dipole auf Impedanz und Abstrahlung“, Funkamateure Heft 4 2026, S. 298ff.
- [3] Englischer Wikipedia-Artikel „Numerical Electromagnetics Code“ [https://en.wikipedia.org/wiki/Numerical\\_Electromagnetics\\_Code](https://en.wikipedia.org/wiki/Numerical_Electromagnetics_Code)
- [4] Open Research Institute (ORI): „Arcanum: Open Source MoM Solver for Antenna Design and Analysis“ <https://github.com/OpenResearchInstitute/Arcanum>
- [5] Tim Molteno et al: NEC2++ Projekt <https://github.com/tmolteno/necpp>
- [6] Andreas Krüger, DJ3EI: „Von Spulen und Kondensatoren zu Filtern und Tunern“, <https://dj3ei.famsik.de/2022-vortrag-hamradio/>
- [7] Andreas Krüger, DJ3EI: antenna-simulation-driver, [https://gitlab.com/4ham/antenna\\_simulation\\_driver](https://gitlab.com/4ham/antenna_simulation_driver)
- [8] Andreas Krüger, DJ3EI: „Antenna simulation driver showcase“ [https://dj3ei.famsik.de/blog/antennas/antenna\\_simulation\\_driver\\_showcase/](https://dj3ei.famsik.de/blog/antennas/antenna_simulation_driver_showcase/)
- [9] B. A. Austin: „An HF multiband wire antenna for single-hop point-to-point applications“ J.IERE, 57, 4, 1987, 167-173, auch [https://www.qsl.net/ab4oj/dl/zs6bkw\\_theory.pdf](https://www.qsl.net/ab4oj/dl/zs6bkw_theory.pdf)
- [10] Brian Austin, G0GSF (ex ZS6BKW): „The ZS6BKW Antenna – from the horse's mouth“ Sprat #130, auch <https://www.wireantennas.co.uk/pdf/zs6bkw-antenna-from-the-horses-mouth-by-g0gsf.pdf>
- [11] SciPy Steering Council: „SciPy, Fundamental algorithms for scientific computing in Python“ <https://scipy.org>
- [12] JS8/JS8Call Homepage: <https://js8call.com>
- [13] Dr. Andreas Krüger, DJ3EI: „Sich mit JS8 anfreunden“, Tagungsband und Vortrag UKW-Tagung 2025, auch <https://dj3ei.famsik.de/2025-JS8/>
- [14] Dr. Andreas Krüger, DJ3EI: „1000 Antennensimulationen“. Hier, sowie auf <https://dj3ei.famsik.de/2026-Antennensimulationen>
- [15] Creative Commons: „CC BY-SA 4.0 Namensnennung-Share Alike 4.0 International“ <https://creativecommons.org/licenses/by-sa/4.0/deed.de>

---

22 Im Nachhinein schäme ich mich etwas, nicht früher Lunte gerochen zu haben. Gier macht blind!