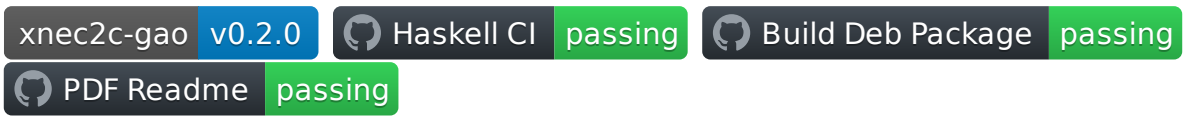
xnec2c-gao – A genetic algorithm optimizer for xnec2c

Maurizio Di Pietro DC1MDP

1. November 2022

Contents

1 Xnec2c-gao



1.1 What is Xnec2c-gao

Xnec2c-gao is an external optimizer for the antenna modeling software Xnec2c written by Neoklis Kyriazis, 5B4AZ. It uses an genetic algorithm to find solutions in a multidimensional search space.

1.1.1 Features

- automatically runs Xnec2c
- symbols in your model

- mathematic expression evaluation
- genetic algorithm control
- unlimited number of genes for genetic optimization
- ability to specify multiple “bands” as optimization targets
- one output nec file per survivor

1.2 Installing xnec2c-gao

1.2.1 Install .deb package

To install

- download the .deb package from release page.
- in your terminal run

```
$ sudo dpkg -i xnec2c-gao_<version>.deb
```

1.2.2 Building from source

To build from source

- download the .tar.gz package from release page.
- install haskell via ghcup
- in your terminal

```
$ tar xzvf xnec2c-gao_<version>.tar.gz
$ cd xnec2c-gao
$ cabal build
$ sudo cabal install
```

1.3 Why do we need an external optimizer for Xnec2c

Xnec2c is a high-performance multi-threaded electromagnetic simulation package to model antenna near- and far-field radiation patterns for Linux and UNIX operating systems. For Windows there are also two main descendants of the nec2 code, EZNEC and 4NEC2, that are cabable of optimizing your antenna model. The author of Xnec2c introduced a optimization loop that uses external optimizing programs.

Neoklis, 5B4AZ explains Antenna Optimization in xnec2c on his page here as follows:

> "Xnec2c monitors its .nec input file for changes and re-runs the frequency stepping loop which recalculates new data and prints to the .csv file. It is therefore possible to arrange the optimizer program to read the .csv data file, recalculate antenna parameters and save them to the .nec input file. xnec2c will then recalculate and save new frequency-dependent data to the .csv file.
>
> If the optimizer program is arranged to monitor changes to the .csv file, then a continuous loop can be created in which new antenna parameters are calculated and saved to the .nec file, new frequency dependent data are calculated and saved to the .csv file and the loop repeated until the desired results (optimization) are obtained."

2 From .nec to .gao file

A .gao file is essentially a .nec file with special “SYM” and “GSYM” cards. With those new cardtypes it is possible to genralize your nec model so the optimizer can generate a whole class of concrete nec models and test them for fitness.

2.1 SYM Card

The SYM Card allows you to bind a expression to a name. After the SYM Card the defined symbol can be used in arithmetic expressions to calculate antenna parameters. A SYM Card is defined as:

```
SYM symbolname := <expr>
```

where <expr> is a mathematical expression consisting of Float/Integer literals, identifiers and operators +,-,*,/,^,(,),SQRT,LOG,SIN,COS.

So you could define the speed of light as :

```
SYM c := 300000
```

and your desired frequency as :

```
SYM freq := 145800
```

you can use previosly defined symbols in the rhs of a SYM Card

```
SYM vf := 0,96
SYM lambda := vf * c / freq
SYM lambdaq := lambda / 4
```

2.1.1 Using SYM Card symbols in your GW Cards

To define a dipole you would define a GW Card for the radiator like :

```
GW    1   15    -0,49   0   3,92   0,49    0   3,92   0,01
```

instead of using concrete values using SYM symbols makes this definition more general:

```
SYM height := 2,0 * lambda
SYM radius := 0,01
GW    1   15    -lamdaq   0   height   lamdaq    0   height   radius
```

It is also possible to use the expression syntax in the GW Cards fields :

```
SYM center := 0,0
SYM height := 2,0 * lambda
SYM radius := 0,01
GW    1   15    center-lamdaq   0   height   center+lamdaq    0   height   radius
```

By itself SYM Cards give you only the ability to adapt your model easily to different circumstances. For example to make the 2m-dipole model a dipole for the 10m band, you would only need to change the SYM freq card to reflect the new 10m band frequency.

```
SYM freq := 28480
```

For xnec2c-gao to be able to optimize this model with an genetic algorithm, we need the ability to define Genes which are implemented as GSYM Cards.

2.2 GSYM Card

The GSYM Card represents essantialy a gene in the genetic algorithm used to optimize your antenna model. You can define any number of GSYM Cards each represents a dimension in a multidimensional search space. Thats why i designed the optimizer as an genetic algorithem , because linear approximation works good for low dimensional search, whereas genetic algorithem have their strength in multidimensional domains.

The Syntax of a GSYM Card is:

```
GSYM symbolname := [<low>...<high>]
```

where <low> and <high> are floating point numbers and low < high.

So to check what radius gives best VSWR/GAIN one would replace the definition of radius from above from:

```
SYM radius := 0,01
```

to

```
GSYM radius := [0,001...0,02]
```

to check radii in the range of 1mm to 2cm.

2.2.1 Using GSYM Card symbols to "modulate" SYM Card symbols

To vary the parameters defined in terms of SYM cards you can use the GSYM symbols in the SYM definition.

```
GSYM  scale := [0,5...1,5]
SYM   lambda := scale * vf * c / freq
```

2.3 BND Cards

In xnec2c frequency ranges are specified as one ore more FR Cards. You can use these and xnec2c-gao will automatically sweep over the bands they define and does calculation of fittnes for all obtained steps. However if for example you specify two FR Cards for antenna model which isn't resonant on both bands, the AVSWR calculation show a distorted reading of the average over data points from both bands, resonant and non resonant. This makes it somewhat harder for the algorithm to rate the variations of the model and hence to produce good results fast.

To prevent this and help the algorithm optimizing your model we have BND Cards in the form :

```
BND <label>   <lowfreq>   <highfreq>    <steps>
```

for 2m ham band it would be:

```
BND 2m  144,000  146,000  10
```

Each BND card instructs xnec2c-gao, to run for each BND Band in isolation for each variant of your model. In presence of BND Cards, FR Cards are not considered in optimization. On the resulting output .nec files, only the original FR Cards are included , BND Cards disapear.

So you could have a FR card that sweeps the entire HF band from 3.5 MHzto 30 MHz and several BND Cards for only the HAM Bands. That would instruct xnec2c-gao to optimize for the HAM Bands and if you run the winner .nec files with xnec2c, it will show the performance for the entire 3.5 Mhz to 30Mhz range.

The fewer and narrower the BND Cards are the faster xnec2c-gao will run. Also the with <steps> specified count of calculation points, will speed up the calculation if <step> is lower. So you could also

define one FR Card that sweeps the 2m band very detailed, and a BND Card for 2m Band that has only a few steps, to make optimization fast and final display detailed.

Example : examples/awx/

3 From .gao file to .gao.nec files

After you prepared your .gao file for your antenna model, you can invoke the optimizer. To get a brief overview how to run the optimizer run:

```
$ xnec2c-gao --help
```

This will yield:

```
xnec2c-gao - a genetic algorithm optimizer for your antenna model

Usage: xnec2c-gao [--version] (-f|--gaofile FILENAME) [-v]
                  [-d|--select-distinct] [-s|--population-size INT]
                  [-c|--generation-count INT] [-o|--optimization-mode omode]
                  [-y|--directional-mode dmode]

  Run an optimizer for GAOModel FILENAME

Available options:
  -h,--help                Show this help text
  --version                Show version
  -f,--gaofile FILENAME    GAO Model to optimize
  -v                       How verbose to optimize (can be specified multiple
                           times)
  -d,--select-distinct     Select only distinct individuals as survivors
  -s,--population-size INT How many individuals are in one generation
                           (default: 20)
  -c,--generation-count INT
                           How many generations to run the optimizer
                           (default: 10)
  -o,--optimization-mode omode
                           What should we optimize for: vswr, gain, vswr+gain
                           (default: vswr+gain)
  -y,--directional-mode dmode
                           Are we optimizing a symmetrical or directive antenna:
                           symmetrical, directive (default: symetrical)

Copyright 2022 Maurizio Di Pietro DC1MDP. Program is provided "as is". Author is
not responsible for any havoc caused by the usage of this software. Use at own
risk.
```

3.1 Running your optimization

So to execute your optimization invoke:

```
$ xnec2c-gao -d -f dipole.gao
```

3.2 Fixing errors in gao file

```
$ xnec2c-gao -d -f dipole.gao
```

This will read your .gao file and report any syntactic errors in the .gao file. With the provided line number and the description what was found and what was expected it should be, more or less, straigthforward to fix errors.

If for example you defined :

SYM vk := 0.96

and run

```
$ xnec2c-gao -f dipole.gao
```

output would be

```
dipole.gao:6:12:
  |
6 | SYM vk := 0.96
  |            ^^
unexpected ".9"
expecting crlf newline, end of input, end of line, or newline
xnec2c-gao: Abort
```

This error doesn't indicate that you used the wrong number delimeter , but at least it indicates where to look.

3.3 Output of xnec2c-gao

xnec2c-gao will run for n generations and present you the evolved survivors, you can choose to continue optimizing or if you are satisfied quit the optimization in which case xnec2c-gao will write one .nec file for each survivor. The filename of the necfile will also contain the AVSWR calculated by xnec2c.

4 TODO

- examples are not clean
- genetic parameters command line options
- sanity checks for radius, segment length, ratios
- verbosity logging
- ...