# S-26.3000 Special work in radio technology

## Graphical antenna measurement software

## User guide

September 8, 2013

Tuomas Leinonen

# Contents

# 1 Introduction

*antMeas* is an easy-to-use graphical antenna measurement software built on top MATLAB, featuring scattering parameter, radiation pattern and efficiency measurements. This software is custom made for the department Radio Sciences and Technology (RAD), Aalto University School of Electrical Engineering as a special work project during the summer of 2013. The software is intended primarily for the smaller anechoic chamber located in the 4th floor of the RAD-wing (i.e., southern C-wing), operated by staff and students alike. Nevertheless, the software can be used for other measurement systems with minor changes.

Some of the key features of this software include, but are not limited to, following:

- Multi-platform, intuitive, easy-to-use, lightweight GUI built on MATLAB version 8.X (2012b and later)
- Command line operable for faster and automated batch processing via scripting, even with older MATLAB versions
- Scattering parameters are visualized on a Smith chart or against frequency using Rectangle coordinates
- Radiation pattern may be plot using rectangular or polar coordinates or as contours or surface, with fully customizable value and axis properties
- Antenna efficiency may be determined easily by measuring an unknown antenna and a reference antenna using any measurement setup, as long as the setup is identical between these two measurements
- Interaction with Orbit software, namely with DataPro+, minimized

In addition to being a user guide, this document duals as a special work documentation. This guide assumes that the user has already measured or otherwise acquired the required "raw" input files. These requirements for different features are listed in the following table (Table 1):

Table 1: Required input file combinations required by different features.

| | File format | | Intended measurement | | |
|---|---|---|---|---|---|
| Required input:[1] | DataPro+[2] | antMeas | $S_{11}$-Params | Pattern[3] | Efficiency |
| AUT S-Parameters | | `*.s#p` | Yes | No | Yes |
| AUT Horizontal | `*.dat` | `*.001` | No | (Yes) | Yes |
| AUT Vertical | `*.dat` | `*.001` | Mo | (Yes) | Yes |
| REF Horizontal | `*.dat` | `*.001` | No | No | Yes |
| REF Vertical | `*.dat` | `*.001` | No | No | Yes |
| REF Efficiency | | `*.txt` | No | No | Yes |

[1] AUT refers to the antenna under testing, while REF is the reference antenna.
[2] `*.dat` files are converted into `*.001` files with DataPro+.
[3] Only one polarization is required for pattern, but both can be used.

In this text, the user is first instructed how to use DataPro+ to convert the raw data in binary coded `*.dat` files from the FR959 measurement system into `*.001` files; a human readable format compatible with antMeas and user's own scripting. In the following section the user is taken through a thorough course into using the graphical user interface (GUI). This part covers a step-by-step guide into analysing an antenna with additional notes and observations about the software that need to be known if a reliable study is desired. In the second to last section, the user is given a deeper insight into the internal structure of the program, aiming for automated batch-processing using command line scripting. The last section is intended for the person that wants to take this project even further, listing some of the numerous possible improvements to antMeas and this guide.

This guide uses the following convention to distinguish between various elements to be interacted with when following this guide on a computer. Roman (i.e., "normal") text does not refer to any text found in the software, but is used for short remarks when companied with one of the following styles that refer to actual elements in the software. *Italian* text is used to refer text or elements of the GUI while `monospaced` text stands for user input (in addition to being used for filenames and URIs). **Bold** text is used to highlight the default option.

# 2 Preparation with DataPro+

DataPro+ can be found on the same computer that is used the run the radiation pattern measurement with Orbit FR959. The PC is located in the third floor of the southern C-wing of the ELEC building. DataPro+ can also be obtained to one's own PC from the RAD server, but there are some compatibility issues with newer PCs. This guide has been written for the aforementioned PC, but for most parts, it applies to every PC running DataPro+.

## 2.1 Running

DataPro+ may be run either by clicking on the desktop icon (see Fig. 1), from the Start-menu (either under *Start > Programs > FR959 Plus v1.1.2 > DataPro Plus Antenna* or under *Start > Programs > 959 Spectrum v2.1 > DataPro Plus Antenna*) or by using the command line (*Start > Run...* or *Start > Run... >* `cmd`) with the following syntax: `C:\Program Files\FR959Plus\datapro.exe -antenna`.

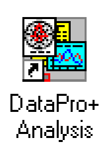


Figure 1: DataPro+ desktop icon

## 2.2 Plot variables

Now that you've got the program running, locate the `*.dat` file containing your measurement using the file browser in the left frame. Double-click on the file to open it and to update the right frame (note that the right frame may look different from the one shown in this guide). Now open the *Plot Variables* window found under *Define* in the menu bar on top of the window. Enable the *Overlay* option under the *SAC Vertical* panel (located in the lower left corner), and click *OK* to close the window. The right frame should update. For these steps, see Fig. 2.

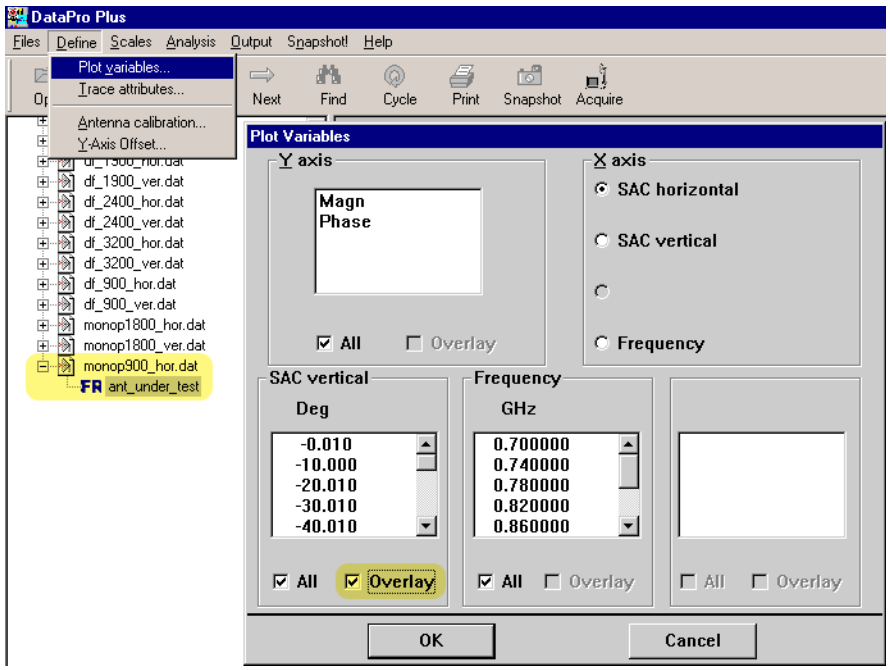


Figure 2: DataPro+ file browser and *Plot Variables* window

## 2.3 Plot definition files

Now it's time to change the output file format to the one suited for antMeas. This is accomplished in the *Load Plot Definition File* window opened by clicking *Files > Load plot definition file...* on the menu bar.

In the newly opened window, the options should be set to equal to those shown in following figure (Fig. 3). The folder open on the right (under *Directories*) should be `C:\Program Files\FR959Plus`. In this folder, one should find `antMeas.pld` using the file browser on the right (labelled *File Name*).

In the *Listing Output* panel (lower), check *Output to Listing File* and *Auto File Name* options. From the drop-down selection on the right (labelled *Output File Name*), choose **.001*. In the field below ("*Output File Path*"), type the destination folder. Hit *OK* to close the window and continue to the next step. The frame on the right should be updated.

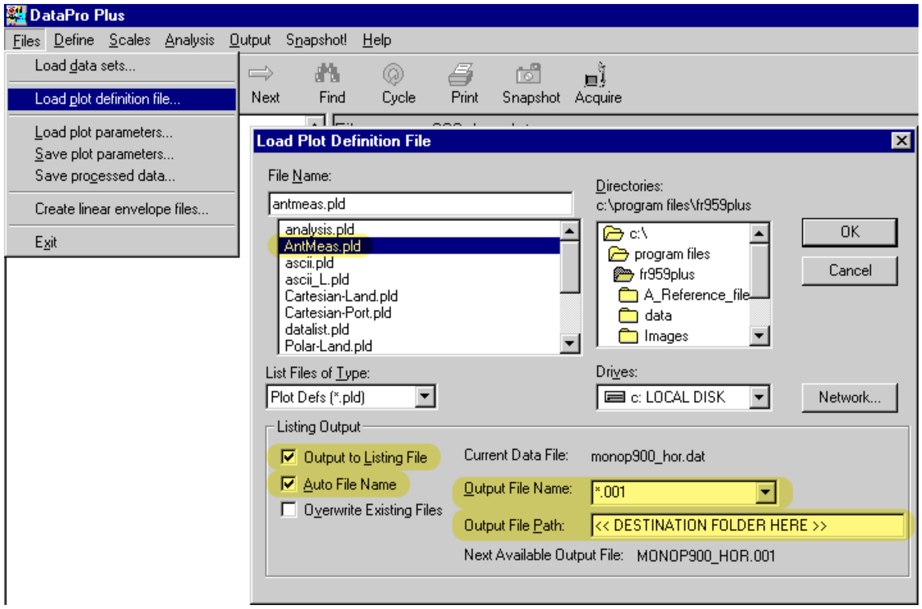


Figure 3: *Load Plot Definition File* window

## 2.4 Exporting

Finally, to save your measurements as `*.001` files, click *Output* > *All defined plots* on the menu bar (see Fig. 4).

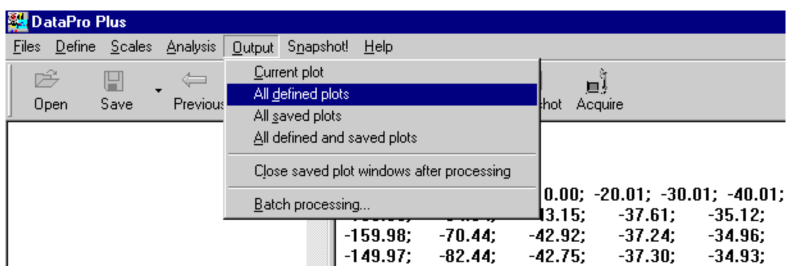


Figure 4: DataPro+ output

Using these steps DataPro creates $2N$ (where $N$ is the number of frequencies) files to your specified folder. All the files have the same name, expect for the extension. The extension increments from each file to next, starting from 001. Files $1 \ldots N$ contain the magnitude information whereas $N{+}1 \ldots 2N$ contain the phase data. Each file should look something like the following snippet:

```
File: monop900_hor.dat
Comment: monop900
Freq:       0.700000 GHz
Prim/Scndry;     -0.01;  -10.00;  -20.01;  -30.01;  -40.01;    ...
  -169.99;   -64.94;   -43.15;   -37.61;   -35.12;   -34.02;   ...
  -159.98;   -70.44;   -42.92;   -37.24;   -34.96;   -33.66;   ...
  -149.97;   -82.44;   -42.75;   -37.30;   -34.93;   -33.80;   ...
  -139.98;  -101.55;   -42.68;   -37.30;   -34.99;   -33.93;   ...
  -129.99;   -72.51;   -42.55;   -37.22;   -34.94;   -33.94;   ...
     ...      ...       ...       ...       ...       ...      ...
```

**Note!** You may use `antMeas.pld` for your own scripts as well. The magnitude and phase are given in dBm:s and degrees, respectively. It supports up to 90 overlays corresponding to a minimum change in SAC Vertical of two degrees.

Repeat these steps for all necessary `*.dat` files (see Table 1) before running antMeas. Since the computer is not connected to the network, you will need to use a “small” (in today’s terms) FAT-formatted USB memory to transfer your files.

# 3 Using the GUI

antMeas core functions are MATLAB version independent, but the GUI elements require version 8.X (that is, 2012b or later). Version is checked automatically when run and an error is displayed if too old version is detected. One can also easily distinguish MATLAB 8.X on Windows from the ribbon-styled toolbar.

The GUI is built using `GUI Layout Toolbox` add-on available from MATLAB Central, since the “official” GUI-elements are quite limited. This package allows for more dynamic graphical interface elements making the GUI more user-friendly. However, since the package is not official, there are some kinks. Nevertheless, the author would like to thank the authors of the used third-party software.

## 3.1 Running

Run MATLAB, locate the antMeas “root” folder (`antMeas.m` file and folders `1st_party` and `3rd_party` should be “directly” visible). Having done this, the user is asked to read the quick-help of antMeas. It can be viewed typing `help antMeas` or `antMeas('help')` into the command window. The help shows how antMeas can be run, and how it’s behaviour may be altered from the default settings.

There are few simple rules for calling antMeas. You may simply type `antMeas` or open `antMeas.m` in the editor and click *Run* to use default settings. You may also pass one additional input argument from the following list using syntax `antMeas('param')`: `'help'` displays help, `'debug'` enables debugging (for advanced users only), `'reset'` clears the current MATLAB instance before running antMeas, and `'nogui'` runs antMeas without the GUI (see next section).

You may also use up to two optional output arguments with following syntax:

```
G = antMeas;                % or
G = antMeas('param');       % or
[G, S] = antMeas;           % or
[G, S] = antMeas('param');
```

Here `G` would contain the `guidata` structure of the program, whereas `S` contains the static data used by antMeas. Output arguments come useful when antMeas is intended to be run, at least to some extent, straight from the command line.

When the software is successfully run with the GUI enabled, it opens to a tab labelled *Input*. The tab is discussed in the next subsection.

## 3.2 Input tab

The input tab (running on Windows 7) is shown in the figure below (Fig. 5). It consists of six file browsing fields divided into two groups of three. The first group, labelled *Antenna under testing* (AUT) contains radiation pattern inputs for both horizontal and vertical polarizations, and an input for scattering parameters. The group labelled *Reference antenna* is equal to the AUT-group with the exception that the last input is for efficiency, not for S-parameters.

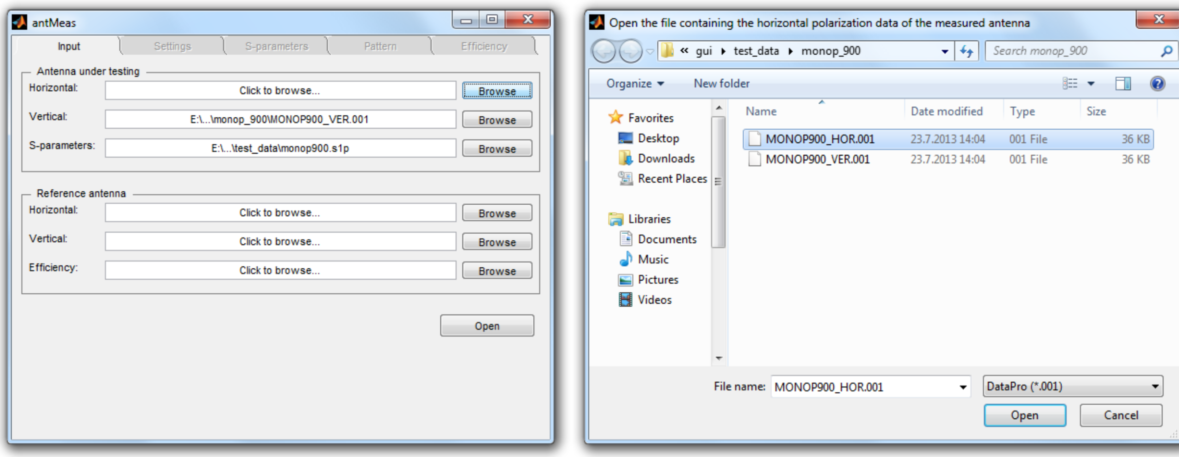


Figure 5: The input tab and a file browsing window

The user needs to define only the inputs needed for the intended analysis according to Table 1. That is, at least one of the AUT-inputs need to be defined. Clicking either on the field or the *Browse* button opens a file browsing window that is used to locate the corresponding input file. An input is defined when the field contains a (shorted) URI of the target file instead of the text "*Click to browse...*". You may redefine an input by clicking either on the field or on the button. You may remove an input by clicking *Cancel* in the file browsing window.

For both antennas (separately), the radiation pattern inputs are linked in the following manner. When browsing for horizontal pattern, the default folder is taken from the vertical input (if defined), and vice versa, making the browsing faster. In other words, the program assumes (but does not enforce) that for any antenna, the pattern files for both polarizations are located in the same folder. Since S-parameters and efficiency data are obtained from other measurements, they are not linked in any way.

When at least one of the AUT-inputs is defined, the *Open* button is enabled. Clicking on the button loads the contents of the defined input files into memory, overwriting any data from previous sessions (if any) as long as new session was started successfully. A session is started when input files are loaded, and multiple sessions may be opened during one run of the program, but not simultaneously. Parallelism is achieved with multiple instances of antMeas.

**Note!** Loading the input files into program memory overwrites any data from previous sessions simultaneously resetting the program into its default state. This does not, however, remove existing figures regardless of their target axis. The docked figures may be unavailable until proper steps are taken in the new session. Also, existing docked figures are obtainable only through command line if the new session does not support those tabs.

Once the input files are successfully loaded, the GUI will switch to the tab labelled *Settings*. The settings tab is covered in detail in the next subsection.

## 3.3 Settings tab

The settings tab is divided into two panels labelled *Global units* and *Interpolation* (see Fig. 6). The global units panel is available with any input combination, enabling the user to change the units of frequency and phase used throughout the program. Frequencies may be given in units ranging from hertz to terahertz, megahertz being the default option. Phase/angle is shown either in degrees (default) or in radians. These options are "dynamic," meaning the changes are saved on the spot and are effective immediately.

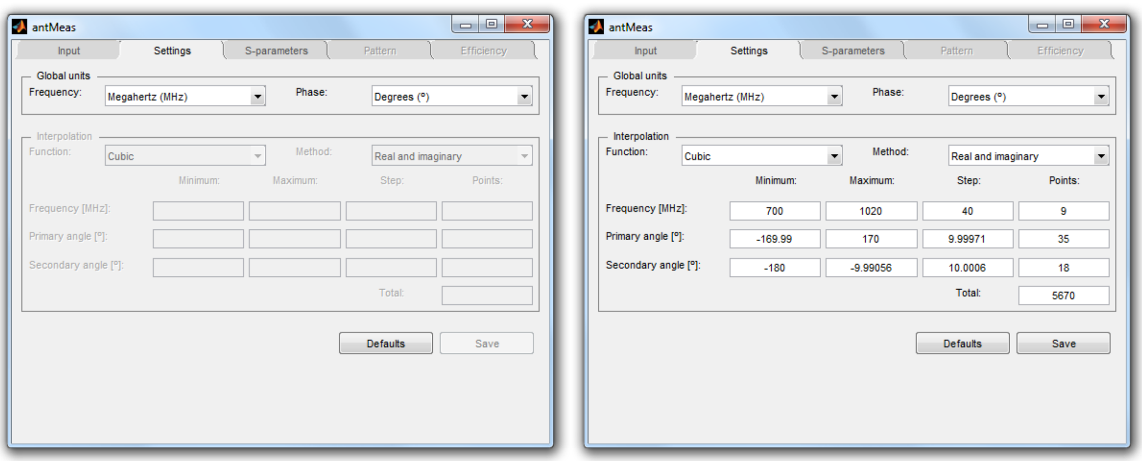


Figure 6: The settings tab when only S-parameters are defined (left), and when at least one radiation pattern is defined (right)

The settings available in the *Interpolation* panel do not affect S-parameters, and are thus available only for pattern and efficiency analyses. The options in this panel include both interpolation function method (labelled *Function* and *Method*, respectively) in addition to the final axis vectors (i.e., frequency, primary angle and secondary angle).

Interpolation function refers to the function that is used to find the interpolated value, and may be defined from the following list: *Nearest* (use the nearest value in the original vector), *Linear* (the value of the straight line connecting adjacent values of the original vector), ***Cubic*** (a third degree polynomial fit passing through all original points) and *Spline* (a less strict version of *Cubic*).

Interpolation method refers to way the two dimensions of complex values are separated for interpolation. The options are as follows: ***Real and imaginary*** (real and imaginary parts are interpolated separately), *Linear magnitude and phase* (the two dimensions are absolute value and the angle), *Logarithmic magnitude and phase* (like the one before, but the magnitude is expressed in decibels).

> **Note!** The three-dimension interpolation is calculated using `meshgrid` and `interp3`. When using cubic interpolation, `interp3` requires all six axis vectors to be equi-spaced. In practice, the primary and secondary angles will not be equally spaced, meaning the `interp3` will actually use a spline interpolant. By default, this change would raise a `'MATLAB:griddedInterpolant:CubicUniformOnlyWarnId'` warning, but it will be disabled for the duration of the interpolation. Also, the function refers to the `'method'` parameter of `interp3`.

The final axis vectors are defined using the table in the lower *Interpolation* panel. All values will be given in the units defined in the *Global units* panel, and are automatically scaled when the corresponding unit is changed. Modifying the limits or the step, scales the *Count* accordingly. Changing the count scales the step to fit the specified amount of equally spaced points between, and including, the limits. You may also extrapolate the axis vectors by a tenth of the corresponding range in either direction. Invalid values are automatically "corrected".

The separate, un-editable *Total* field is the number of complex points in an interpolated pattern matrix. It aims to give the user some insight into the workload faced by the program. The total number of interpolation points $N_{\text{total}}$ equals to:

$$N_{\text{total}} = \underbrace{(1 \veebar 2 \veebar 4) \times 2 \times N_{\text{freq}} \times N_{\text{prim}} \times N_{\text{sec}}}_{\text{Pattern}} + \underbrace{(0 \veebar 1) \times 2 \times N_{\text{freq}}}_{\text{S-parameters}} + \underbrace{(0 \veebar 1) \times N_{\text{freq}}}_{\text{Efficiency}}$$

Clicking on the *Save* button performs the interpolation, and enables either only the *Pattern* tab, or both *Pattern* and *Efficiency* tabs, depending on the defined inputs. The tab is also changed upon saving. These tabs are explained in the following subsections. The user may change their interpolation settings by saving them again, overwriting any previously saved/interpolated data.

Clicking on the *Defaults* button reverts the tab to its default state with original, unaltered values. While clicking this button does not affect the previously interpolated data, it does reset the global units that are effective immediately.

> **Note!** Since in practice each of the measurements will have different axis vector values, the original, unaltered values are acquired using a "pessimistic min-max & max-min" process. Each of the original axis vectors is constructed from the range present in all the measurements, using the greatest step in those measurements. This is done to avoid giving the user a false illusion about the amount of original data.

## 3.4 S-parameters tab

The S-parameters tab offers the user means to visualize scattering parameters as a function of frequency from various choices when the input is defined. This tab is the first plotting tab, and like all plotting tabs, it consists of two panels: *Plot settings* and *Graph*, and their roles are quite simple: the latter may be used as a "canvas" using the controls in the former.

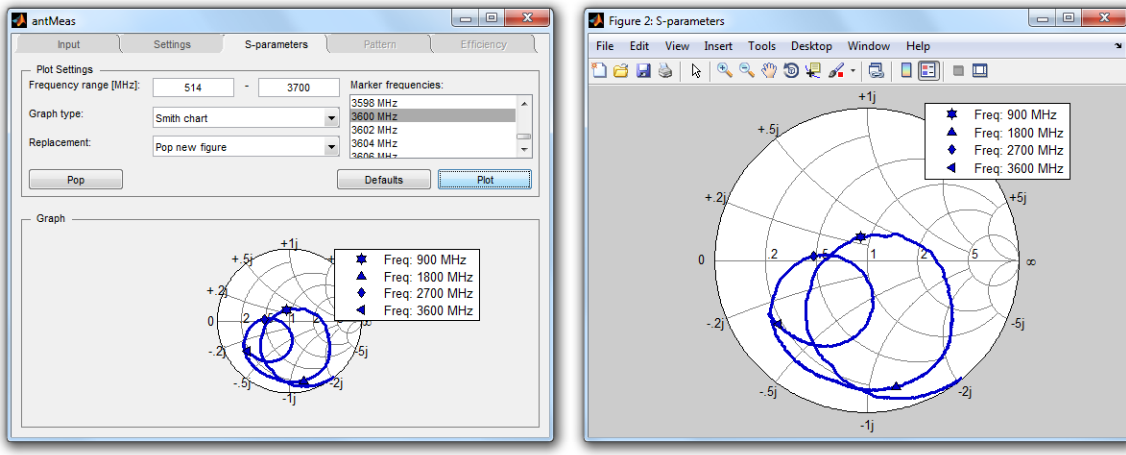


Figure 7: The S-parameters tab and a Smith Chart

The plotting range is defined with the two fields labelled *Frequency range [_Hz]*, one for each limit. You may use the whole range, or crop some points out. You may even use just one frequency point, but you may not extrapolate since S-parameters are not interpolated. Again, all invalid input is "corrected" automatically. This is due to the ease in their measurement: scattering parameters are really easy (and fast) to measure even with large number of points.

Modifying the frequency range updates the *Marker frequencies* list on the right. The user may select any number of frequencies for optional markers. Markers are most commonly used with a Smith chart to highlight points corresponding to certain frequencies. While they provide means to add frequency "axis" to a Smith chart, they may as well be used for other types of plots, where they provide decibels-to-linear and linear-to-decibels conversions as extra info.

> **Note!** While there are no hard-coded limits for marker frequencies, using too many markers makes the graph unreadable.

The *Graph type* selection defines the type of the plot. Offered types are: ***Smith chart*** (complex data, frequency only via markers), *Linear magnitude*, *Logarithmic magnitude*, *Phase*, *Linear magnitude and phase* and *Logarithmic magnitude and phase*. The last two options are so-called dual-plots.

**Note!** Dual-plots are incompatible with the docked axes due to limitations in the third party software. When a dual-plot is selected, the *Replacement* select is set to *Pop new figure* and disabled.

**Note!** It's recommended that Smith charts are used in a separate window, either via popping or via target axes, since one may then use data tips to display values along the curve.

The other selection, labelled *Replacement*, is used to choose the target axis. The choices are: *Replace docked* (use the initially hidden axis located in the *Graph* panel, overwriting any existing figure) and *Pop new figure* (creates a new, stand-alone figure window).

The graph is created when the *Plot* button is clicked. The *Pop* button is enabled when a docked figure is plotted for the first time, allowing the user to copy the docked graph into a new figure. The *Defaults* button works similarly to the one in the *Settings* tab. The *Replacement* selection, and the buttons (*Plot*, *Defaults* and *Pop*) work similarly in all plotting tabs.

**Note!** There are some known scaling issues (due to the third-party extensions) with docked graphs when resizing the window. These are resolved by re-plotting after resizing. The user is encouraged to maximize the window before plotting to get the most out of the docked graphs.

**Note!** The user may save their plots for later when it's plotted in an external figure window. While it is indeed possible to copy an existing docked figure into a new window, the user is advised to avoid this if quality plots are desired. These are achieved by plotting graphs into a new window straight away. These known issues are caused by the third-party software.

## 3.5 Pattern tab

The Pattern tab is similar to the S-parameters tab, but for plotting the radiation pattern. The behaviour of common elements, such as the buttons and the *Replacement* select, is identical to those discussed earlier. A screenshot of the tab is shown in Fig. 8.

The *Graph type* select offers following options: ***Rectangular v = v(x)*** [plot with rectangular coordinates], *Polar v = v(x)* [value as radius respect to an angle], *Filled Contour v = v(x, y)* [coloured contour plot] and *Surface v = v(x, y)* [colour-for-value surface].

1D plots refer to graphs where the value is plot on one axis against another axis, whereas 2D plots is used a general term to describe plots where the value is not shown

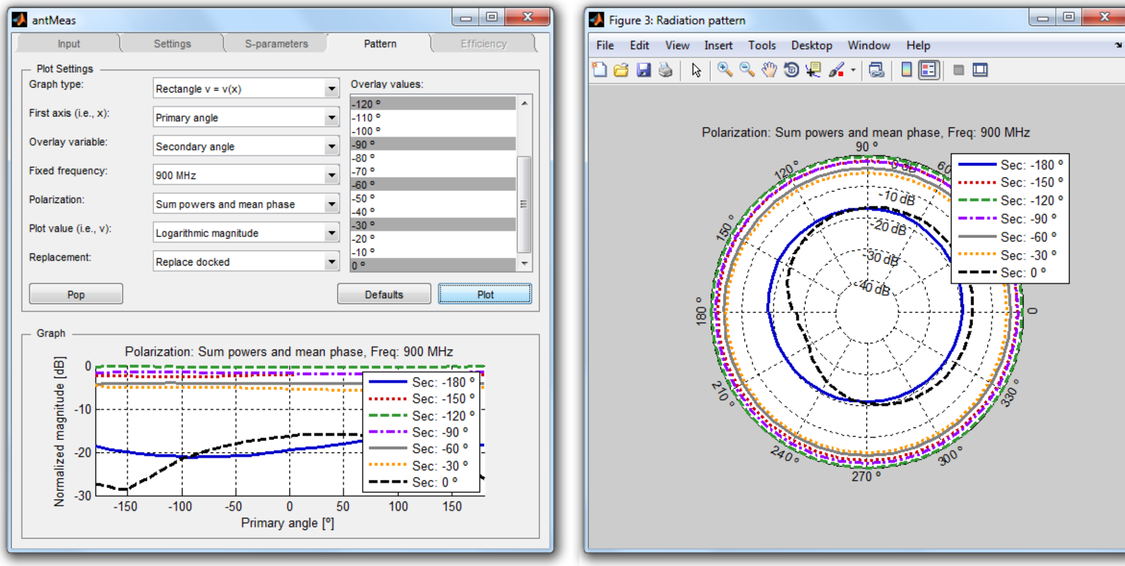


Figure 8: The Pattern tab with a docked rectangular plot and the same plot in a separate window using polar coordinates.

using any conventional axis, but with a colour, for example. In the previous list, the first two choices are 1D plots, and the latter two are 2D plots.

**Note!** The polar plot does not scale automatically, but uses a set of predefined limits depending on the plot value. These limits are $0 \ldots 1$ for linear magnitude, $-50 \ldots 0$ dB for logarithmic magnitude and $-180 \ldots 180\,°$ for phase.

**Note!** Surface may not be plot on the docked axis. This is due to the limitations in the third-party software.

The *First axis (i.e., x)* select defines the first, i.e., horizontal, axis to be used when plotting. The choices are the fixed parameters of the measurement setup: primary angle (default), secondary angle and frequency.

**Note!** Frequency cannot be used for the angular axis when using polar plots, and is not available in the selection. If secondary angle is used for the angular axis, only half a circle is shown. The axis values are rotated to fit inside the shown half ($0 \ldots 180\,°$). Thus the absolute values have even less a meaning they had before.

The select below *First axis (i.e., x)* can have two different labels depending on the graph type. When two-dimensional plots are used, two axis are available and the label reads *Second axis (i.e., y)*. In this case, the select is used to define the variable used for vertical axis. Offered options are the remaining two fixed parameters of the measurement setup, since one is already in use.

When using 1D plots, the second axis may be simulated with overlays. That is, the user may plot multiple graphs in the same figure with different values of the "second axis," called overlays. Now the label reads *Overlay variable*, but the select functions the same way as before.

The number of curves, and the actual values of the overlay variable are controlled with list on the right (labelled *Overlay values*). The user may define any number of graphs (at least one) that are eventually plot with different styles and corresponding legend entries. See Figure 8 for an example. The list is greyed-out, i.e., disabled, for two-dimensional plots.

Also the fourth selection from the top has a dynamic label of the format *Fixed _*, where _ is replaced with the only unused axis parameter. The options are the interpolated values of the corresponding parameter using global units. Only one fixed parameter may be defined in each plot, meaning multiple plots are required if all three parameters are to be swept simultaneously.

The *Polarization* select enables the user to choose between different measurements, or one of their combinations. If both polarizations are available, the options are *Horizontal*, *Vertical*, ***Sum powers and mean phase*** (used for efficiency calculation) and *Sum fields*. When only one of the polarizations are available, the defined polarization is automatically selected and the selection is disabled (and shown as greyed-out).

**Note!** Since a coherent measurement setup is not available, the use of *Sum fields* option is quite absurd. Also, the phase information with the *Sum powers and mean phase* option has no physical meaning. They are provided only for completeness (and for future releases).

The last of the new/unfamiliar controls is the *Plot value (i.e., v)* selection. If offers the user with following options: ***Linear magnitude***, *Logarithmic magnitude*, *Phase*, *Linear magnitude and phase* and *Logarithmic magnitude and phase*. The last two options are so-called dual-plots that are always plot in a new figure.

## 3.6 Efficiency tab

Unlike other plotting tabs, the *Efficiency* tab is divided into three panels. In addition to the familiar *Plot settings* and *Graph* panels, there is a panel labelled *Export* (see Fig. 9). Clicking the *Export* button inside the panel will open a file browsing window. This window is used to define the target file into which the efficiency data is saved. After exporting, the previously unknown antenna may be used as a reference antenna.

In the *Plot settings* tab, there is only one selection in addition to the previously introduced *Replacement*. The *Unit* select offers the user a choice between typical units to express efficiency. These are ***Percent [%]*** $(0 \ldots 100)$, *Decimal* $(0 \ldots 1)$ and *Decibels* $(-\infty \ldots 0)$. Other elements work as previously mentioned.

The efficiency graph always comprises three curves: one for matching efficiency, one for radiation efficiency and one for total efficiency. The total efficiency is calculated once before the first plot or export, and the radiation efficiency is obtained from it with the

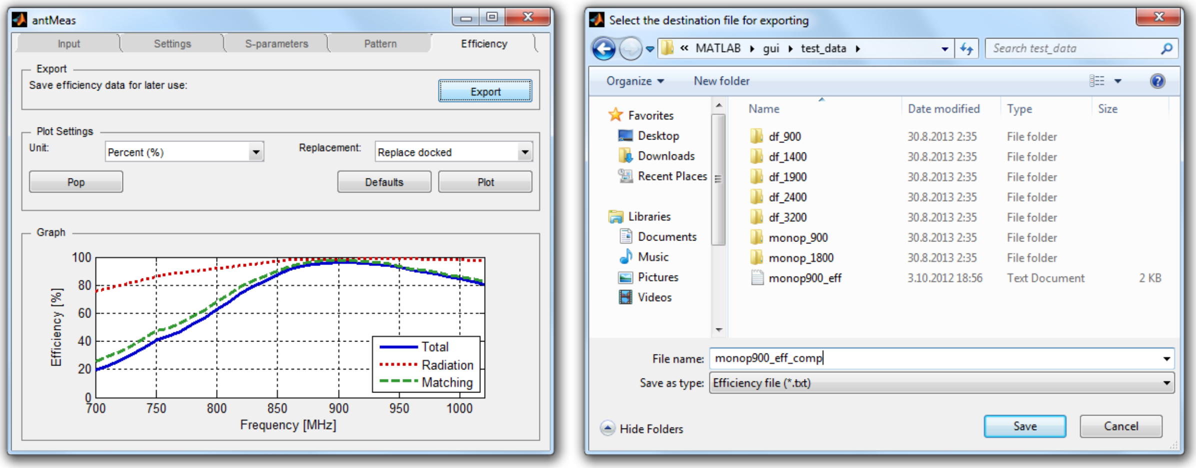


Figure 9: The Efficiency tab with an efficiency plot on the docked axis, and a file browsing window.

help of the absolute value of S-parameters. The graph is always similar with the exception of the unit used for vertical axis.

> **Note!** The axis vectors defined in the *Interpolation* panel of the settings tab are used in the efficiency calculation, allowing the user to find the efficiency when radiating to specific direction. While this software uses "total efficiency" to refer to any range, it is used more commonly to refer to the case where a full sphere is analysed.

# 4 Running from the command line

The only way to master the program is to get acquainted to the actual program code. Even though comments improve clarity, it can be quite a challenge to tackle the code. The following two subsections aim to lower this mental step when a high degree of automation is intended. The first subsection covers the way information is stored within the program, and callbacks are discussed in the latter.

## 4.1 Data structures

The program data is divided into two groups: static (i.e., constant) and dynamic (i.e., variable) data. Both of these have their own `struct`s that are initialized with `initStatic` and `initDynamic(static)`, respectively. How the data is arranged as a tree into these structures is discussed next, starting with the static data. The reader is encouraged to open the `*.m` files in editor for maximum effect. As a general rule it can be said that the data is quite fine-grained into smaller substructures.

The first level offers following sub`struct`s: `window` for main window settings, `general` containing GUI styles and geometry, `tabs` for tab-specific settings and `plotting` to group visualization styles. The `tabs` is divided into general settings and the actual tabs, which are in turn grouped into panels for literals and "purpose-groups" for other elements (`uicontrols`) based on the GUI. Label literals are separated from those that relate to user input. While this grouping causes some literals to appear multiple times for increased memory usage, it also allows for more versatile customization.

> **Note!** All literals are obtained from the `readString` function allowing antMeas to be completely translated into another language. See `initStatic.m` and `readString.m` for more information.

The `gui` is a `guidata` structure containing a handle to the main window (field named `window`), debugging mode (accessed with `gui.debug`), dynamic data (`dynamic` substructure) and the actual interface: `_Tab` structures for tabs and `mainLayout` for tabbing interface. The tab-specific substructures are organised based on the highest-level container they appear in (most commonly a panel). Before and after any changes are made, the struct needs to be refreshed using the following syntax:

```
gui = guidata(gui.window);      % load
%   You code goes here...
guidata(gui.window, gui);       % save
```

The dynamic data tree is divided into sub trees by their purpose since most of the data is accessed in multiple tabs. For example, there are substructs for original data (`.original`), interpolated data (`.interpolated`) and the I/O settings (`.io`) in addition with some general-purpose data. However, plot settings are stored into tab-specific substructures.

> **Note!** After initialization, the dynamic data structure will contain all required fields. That is, all references are valid from the very beginning. When possible, the struct will be initialized from the static data. Other fields are initialized with a "meaningful value," if available. The fields with no obvious default values are left empty (`[]` for numeric data and `''` for text).

## 4.2 Safe callbacks

All stand-alone functions (found under `1st_party\other\`) are safe to be called without the GUI, and can be used as a part of one's own scripts. While some of the callbacks are compatible with the command line version, most of them are not. The following list points out the callbacks (located under `1st_party\gui\callbacks\`) that are compatible with the `'nogui'` parameter:

- `inputLoadFiles` (load data into memory)
- `settingsSave` (interpolate)
- `settingsSave` (update dynamic elements)
- `scattPlot` (plot S-parameters)
- `pattPlot` (visualize radiation pattern)
- `effPlot` (create efficiency graph)
- `effExport` (export results)

> **Note!** Functions listed above still require a figure for `guidata`, since they are not stand-alone. There may be some restrictions concerning the settings in `gui.dynamic`. For example, the inexistent docked axes cannot be used when plotting. And do remember to pass an empty matrix `[]` as the first parameter for non-GUI behaviour.

# 5 For future releases

The following list includes some useful features that were dropped out from the release due to limited resources:

- Polarization measurement (requires a coherent setup)
- Full 3D rotation of the pattern data (mandates a fixed "origin")
- User guide: a FAQ-section (needs user feedback) and other improvements
- Improved sessions (saving/loading/parallelism) and commenting
- More advanced and automated scaling
- Workaround to get rid of some limitations imposed by the third-party software
- Optimizations