The Strak Machine

Short Instruction

Matthias Boese V1.0, 21.09.2020

This short instruction will explain some basic functions of the Strak Machine that are mandatory for using it.

Please take a look at „Strak_Machine_Reference" for the full reference of functions that are included in the Strak Machine.

1. Main purpose / function of the Strak Machine

For aerodynamic reasons, the wing of an aircraft very often does not have a rectangular-shape with a constant chord-length all along the wingspan.

When the chord-length changes along the wingspan at a fixed airspeed, there will be a change in Reynolds-numbers, that is proportional to the change in chord-length.

If the same airfoil is used all along a wing with changing chord-length, this will lead to a loss in performance of the wing, as the airfoil will only have the optimum performance at a certain position-range along the wing-span.

The Strak Machine can help to improve the overall wing-performance by creating a set of airfoils, that are optimized according to the different Reynolds-numbers that occur along the wing-span.

The user specifies a set of Reynolds-numbers for the root-airfoil and for the following airfoils, that will be passed to the Strak Machine by a configuration-file.

Each Reynolds-number, that is specified there (except for the Reynolds-number of the root-airfoil, which is the first), will lead to a new airfoil to be created and that is optimized for this Reynolds-number.

In order to get the parameters to control the optimizer (which is Xoptfoil-JX) for creating the new airfoil, the root-airfoil will first be analyzed at the given Reynolds-numbers, using Xfoil to carry out polar-calculations.

The polars of the root-airfoil will be imported by the Strak Machine and a set of proposed target-polars for the new airfoils will be calculated.

The user can apply trim-values to each single target-polar to adapt the shape of the polar to his needs.

The target-polars will then be used to generate input-files for Xoptfoil-JX, that will carry out the desired optimizations and will create the new airfoils.

2. Two different Versions: „Strak Machine pure“ / „Strak Machine instant“

As the current implementation of the Strak Machine is based on Python 3.x , usually a Python-Distribution and additional packages have to be installed in order to use it.

Also several paths (e.g. for the python-interpreter) have to be set and in some cases the windows-registry has to be patched, so that the command-line-arguments will be properly passed towards the python-interpreter.

Additional packages, like matplotlib, numpy, a Fortran-namelist-parser etc. have to be installed.

To improve „ease-of-use“ for those users, who do not have much experience with python, there will be a precompiled version of the Strak Machine (using „pyinstaller“), that will can be used without having to install python.

This first version is called „Strak Machine instant“.
The Download-size of this version will be larger, as all necessary Python-ressources will be included in the „compiled“ .exe-version.
This version also is portable and could even be started from an USB-stick.

It should be mentioned here, that the starting time for this version is a little bit longer, because the .exe-files will be automatically unzipped after starting them.

The second version „Strak Machine pure“ is intended for advanced users, who either already have a python-installation or do not shy away from installing it, know how to install additional packages and so can take advantage of the smaller download size.

Both versions feature the same functions.

3. Getting Started

Before getting started, in case of „Strak Machine pure“ you may have to install Python.
You can follow the steps 3.1.. 3.4 to complete the installation**.**
In case of „Strak Machine instant“ you can directly jump to step 3.5.

3.1 Strak-Machine pure: installing a Python-Distribution

Strak machine pure requires a Python 3.x-Installation.

The installation can either be downloaded from https://www.python.org/downloads/
or you can use „Python-Anaconda“, which is a more extensive distribution that already includes the package „matplotlib“, which will be necessary for the Strak Machine.

3.2 Strak-Machine pure: Installing additional packages

After you have installed Python 3.x, you have to install the following additional packages using the „pip install“ command in the windows commandline:

```
pip install f90nml
pip install colorama
pip install termcolor
```

If you did not choose „Anaconda“, but a smaller Python-Distribution, you will also have to execute:
pip install matplotlib

Note: In some python-versions a specific version of matplotlib is required, as there are dpendencies between the packages.

There is a specific pip-command to install a certain version of a package (hint: type „pip –help“ to get all features of pip).

3.3 Strak-Machine pure: patching the windows registry

You may have to patch the windows registry in order to get the commandline-arguments correctly passed to the python-interpreter.

To do this, open „RegEdit“ and navigate to HKEY_CLASSES_ROOT\Applications\python.exe\shell\open\command.

Right click on name (Default) and Modify. Enter (for Example):

```
"C:\ProgramData\Anaconda3\python.exe" "%1" %*
```

The red characters above show you, what is probably missing in the existing registry-key.

3.4 Strak-Machine pure: setting the path environment-variable

The path to the Python-Interpreter, that you have installed has to be included in the „path“-environment-variable, so you have to add it there (if it has not automatically been added before during python-installation)

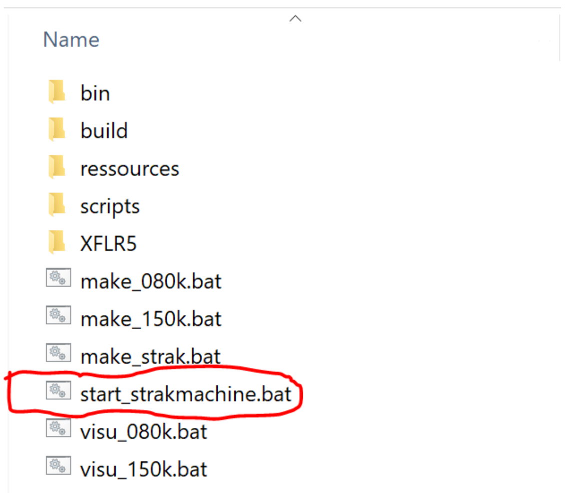
3.5 Both versions of Strak Machine: unzip the archive

Unzip the zip-archive „strakmachine_instant.zip“ or strakmachine_pure.zip“ to your desired location.

Inside the unzipped folder you will find a complete runnable version that is setup with an example „SD-strak“.

3.6 Starting the Strak Machine

Now you are ready to take a first look.
Start the batchfile „start_strakmachine.bat“:



Name

- bin
- build
- ressources
- scripts
- XFLR5
- make_080k.bat
- make_150k.bat
- make_strak.bat
- start_strakmachine.bat
- visu_080k.bat
- visu_150k.bat

If every previous step was carried out correctly, the next thing you should see is a window showing the analysis of the „SD-airfoil“:

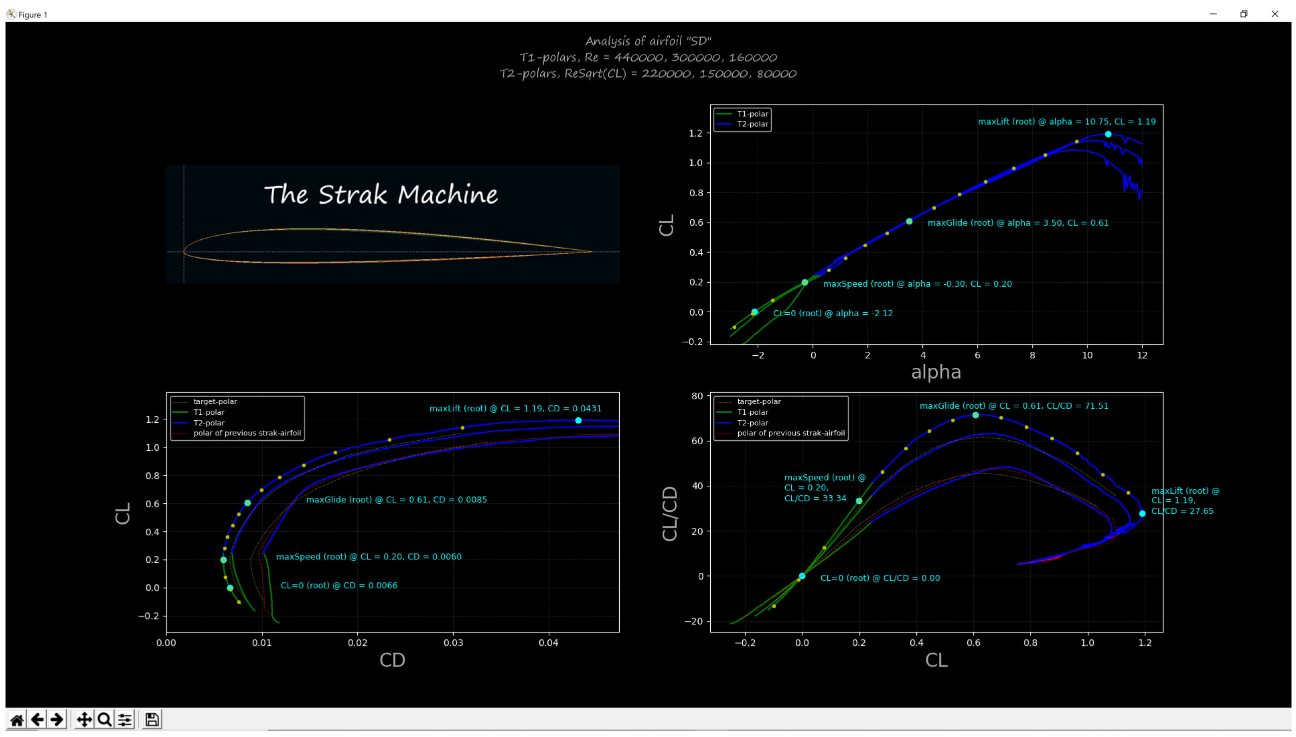
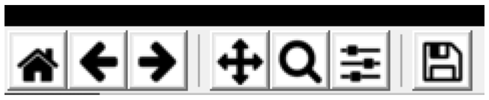


Figure 1
Analysis of airfoil "SD"
T1-polars, Re = 440000, 300000, 160000
T2-polars, ReSqrt(CL) = 220000, 150000, 80000
The Strak Machine
T1-polar
T2-polar
maxLift (root) @ alpha = 10.75, CL = 1.19
maxGlide (root) @ alpha = 3.50, CL = 0.61
maxSpeed (root) @ alpha = -0.30, CL = 0.20
CL=0 (root) @ alpha = -2.12
CL
alpha
target-polar
polar of previous strak-airfoil
maxLift (root) @ CL = 1.19, CD = 0.0431
maxGlide (root) @ CL = 0.61, CD = 0.0085
maxSpeed (root) @ CL = 0.20, CD = 0.0060
CL=0 (root) @ CD = 0.0066
CD
maxGlide (root) @ CL = 0.61, CL/CD = 71.51
maxSpeed (root) @ CL = 0.20, CL/CD = 33.34
maxLift (root) @ CL = 1.19, CL/CD = 27.65
CL=0 (root) @ CL/CD = 0.00
CL/CD

The main-window will be divided into four sub-windows.

You should see the strak-machine-logo in the upper left sub-window and three polar-graphs in the other sub-windows.

You will also find a small toolbar that allows navigation like „zooming“ and „scrolling“ in each of the sub-windows and you also can save an image of the main window to disk.



Learning by doing is probably the best here, to understand it.

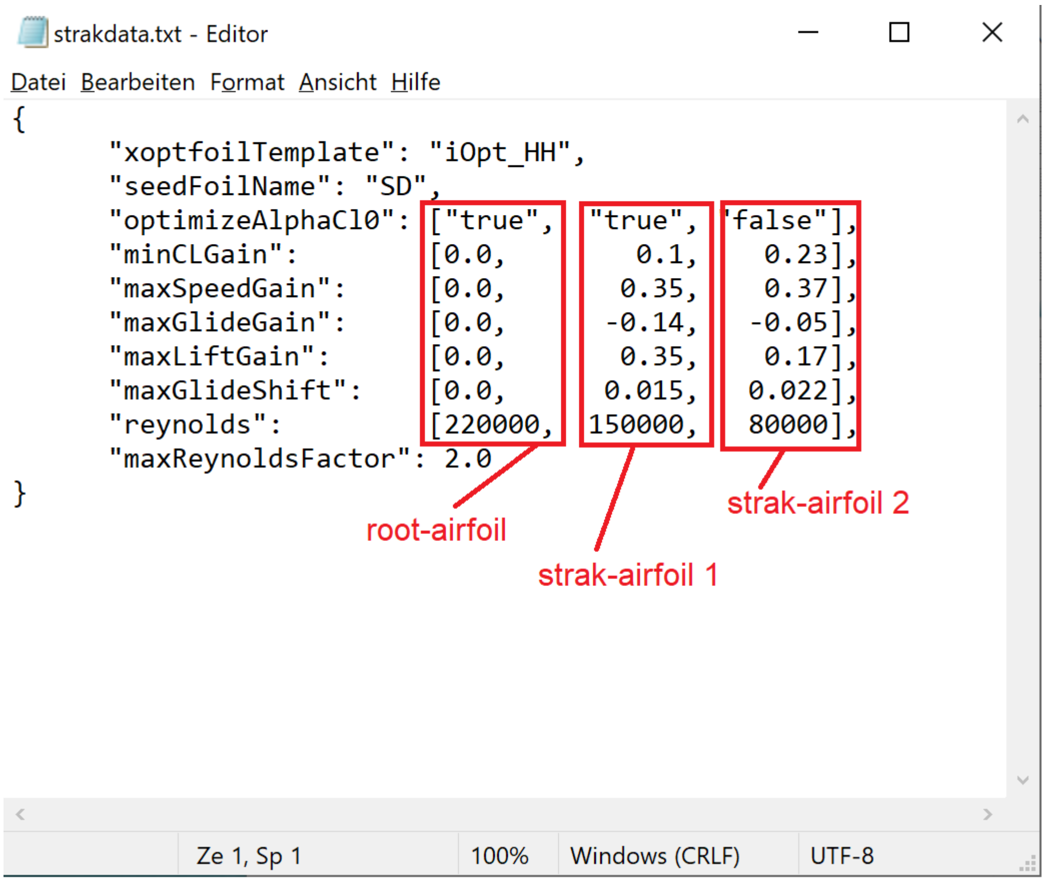
4. The configuration-file „strakdata.txt“

After the start, the Strak Machine will first of all automatically read the file „strakdata.txt“ from the „ressources“-folder.

This is an ASCII-textfile in so called „json“-format, that contains configuration-information for the Strak Machine, telling it for example which airfoil to take as a root-airfoil and which polar-calculations to carry out.

This file has to be adapted by the user for his own project.

When you open the file in a text-editor, you should see the following lines (without the red additions of course, which have only been added for explanatory purposes):



```
{
        "xoptfoilTemplate": "iOpt_HH",
        "seedFoilName": "SD",
        "optimizeAlphaCl0": ["true",  "true", "false"],
        "minCLGain":        [0.0,      0.1,    0.23],
        "maxSpeedGain":     [0.0,      0.35,   0.37],
        "maxGlideGain":     [0.0,     -0.14,  -0.05],
        "maxLiftGain":      [0.0,      0.35,   0.17],
        "maxGlideShift":    [0.0,     0.015,  0.022],
        "reynolds":         [220000, 150000,  80000],
        "maxReynoldsFactor": 2.0
}
```

This example file of „strakdata.txt“ does not contain the complete set of parameters of the Strak Machine, but only mandatory parameters that must be set by the user.

Further parameters can be added by on demand.

The other optional parameters, that are not shown here, will be used with internal default values and they will be described in the document „Strak_Machine_Reference“.

The order of the parameters / sorting of the lines can be changed without having an effect on the function of the Strak Machine (if you would like to have a different sorting, you may do that).

Now follows an explanation of the mandatory parameters and how to use them:

„xoptfoilTemplate“
The name of the template-file, that will be used in order to generate the input-files for Xoptfoil that are necessary for the airfoil-generation.
The name must be written without the ending „.txt“.
The file can be found in the „ressources“ folder.
In this example case, a template file will be used to carry out the airfoil-optimization using „Hicks-Henne“ functions.
The advanced user also can provide his own template file here.
The generated input-files later can be found in the „build“-folder, all named “istrak_...k.txt“.

„seedFoilName“
The name of the airfoil-file that will be used as the root-airfoil
The name must be written without the ending „.dat“.
The airfoil-data has to be provided as an ASCII-textfile in selig-format and must be located at the „ressources“ folder.

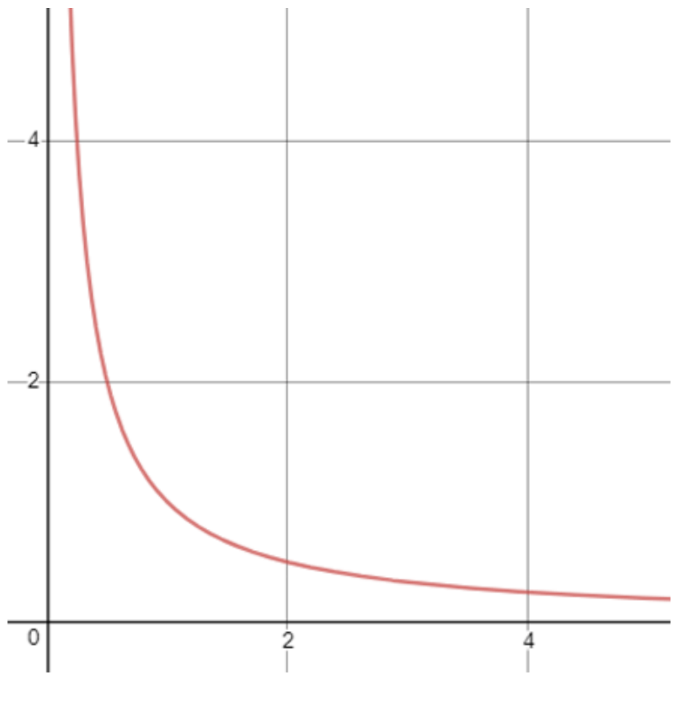
„reynolds“
A list of numbers that represent **ReSqrt(C_L)-values for a type 2 polar** calculation, the number of list-elements equals the overall number of airfoils.

The first number belongs to the root-airfoil, the following numbers belong to the following airfoils that shall be created.

For a correct function of the strak Machine the numbers must be sorted in descending order (that may be improved in upcoming versions, but is a restriction today).

The Strak Machine partly works with „type 2“- (fixed lift) polars.
This means, that the number that is entered here, will be represent Re@C_L = 1.0 and Re will be implicitly calculated / changed when C_L changes.

The function Re(C_L) is hyperbolic, that means when C_L gets smaller, Re will get infinite, the graph shows an example:



To keep Re limited to values, that will probably occur in reality during flight and not getting too large or even infinite, the Strak Machine will set a limit, that will be determined from „reynolds“ and „maxReynoldsFactor“.

„maxReynoldsFactor“
This factor will be used to limit the maximum Reynolds-number that is used for the analysis of the airfoil.

The Strak-machine will always carry out two polar-calculations for each number that is specified in the „reynolds“-list.

1. Type 2-polar-calculation with ReSqrt(C_L) = number specified in the „reynolds“-list
2. Type 1-polar-calculation with Re = maxReynoldsFactor ***** number specified in the „reynolds“-list

In the example-file, the value of „maxReynoldsFactor“ is set to 2.0, and ReSqrt(C_L) for the root-airfoil is set to 220000.

So the Strak Machine will perform a Type 2 polar calculation with ReSqrt(C_L) = 220000 and a Type 1 polar calculation with Re = 440000.

For the analysis of the airfoil, these two polars will be merged automatically by the Strak Machine to a mixed Type 1 / Type 2 polar that will be displayed in the three graphs in the main window.

The change in colour shows, where the polar changes from Type 1 to Type 2.

The switching between the two polar-types will occur at a certain C_L, ‚that will be automatically calculated from „maxReynoldsFactor“ (merging-point).

„optimizeAlphaCL0“
This parameter tells the Strak Machine, whether to carry out an optimization for the zero lift angle (try to keep the same zero lift angle for the airfoil as the root airfoil has).

The zero lift angle is the angle of attack of a wing profile against the airflow, where neither lift nor downforce is generated.

The zero lift angle also strongly determines the maximum lift.

When the difference in Re between root-airfoil and the following airfoils increases, it may be necessary to switch off this optimization or otherwise the generated airfoil gets too much deformed.

It is recommended to keep this optimization activated along the wing from root to tip as long as possible.

The „trimming“ parameters ...Gain and ...Shift

The next five parameters, that will be described here, are used for trimming the „target- polars“, that are calculated by the Strak Machine.

The concept of the Strak Machine is, to setup a target-polar for each of the following-airfoils that shall be created.

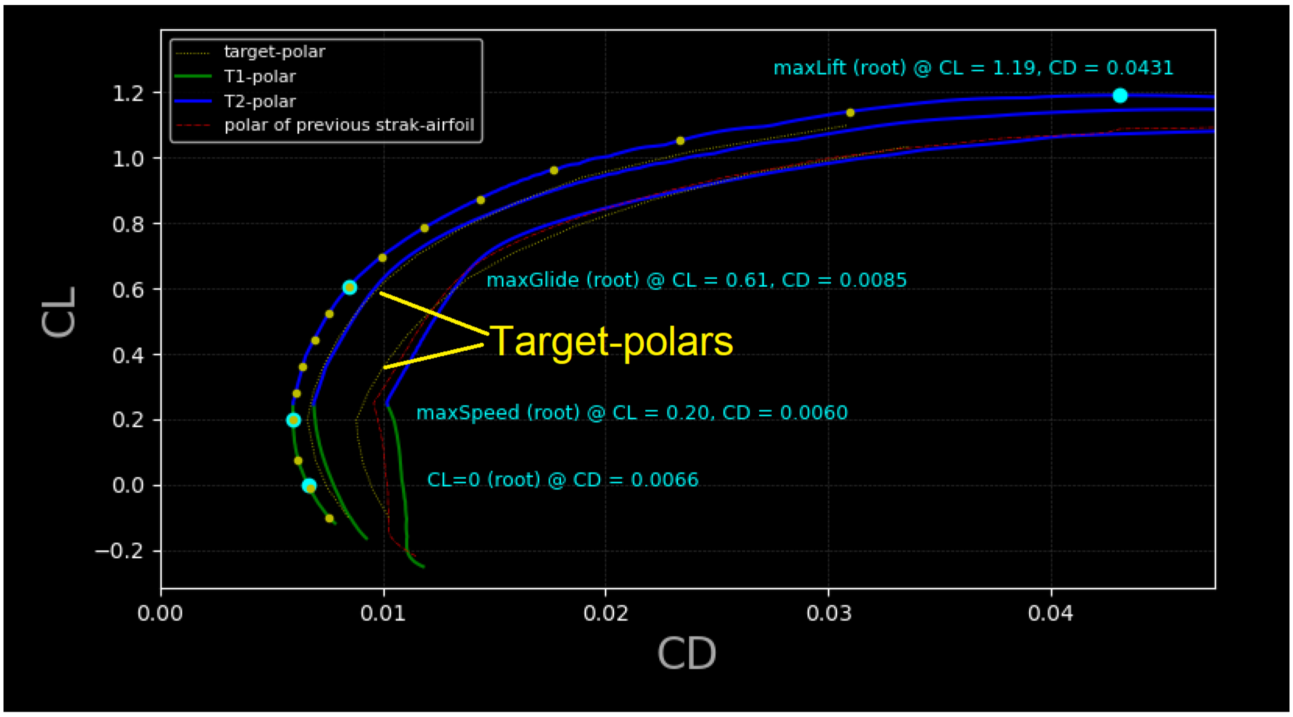
The target-polar-information will be passed to Xoptfoil-JX by a parameter-file for each airfoil and the airfoil will be optimized to match the target-polar information.

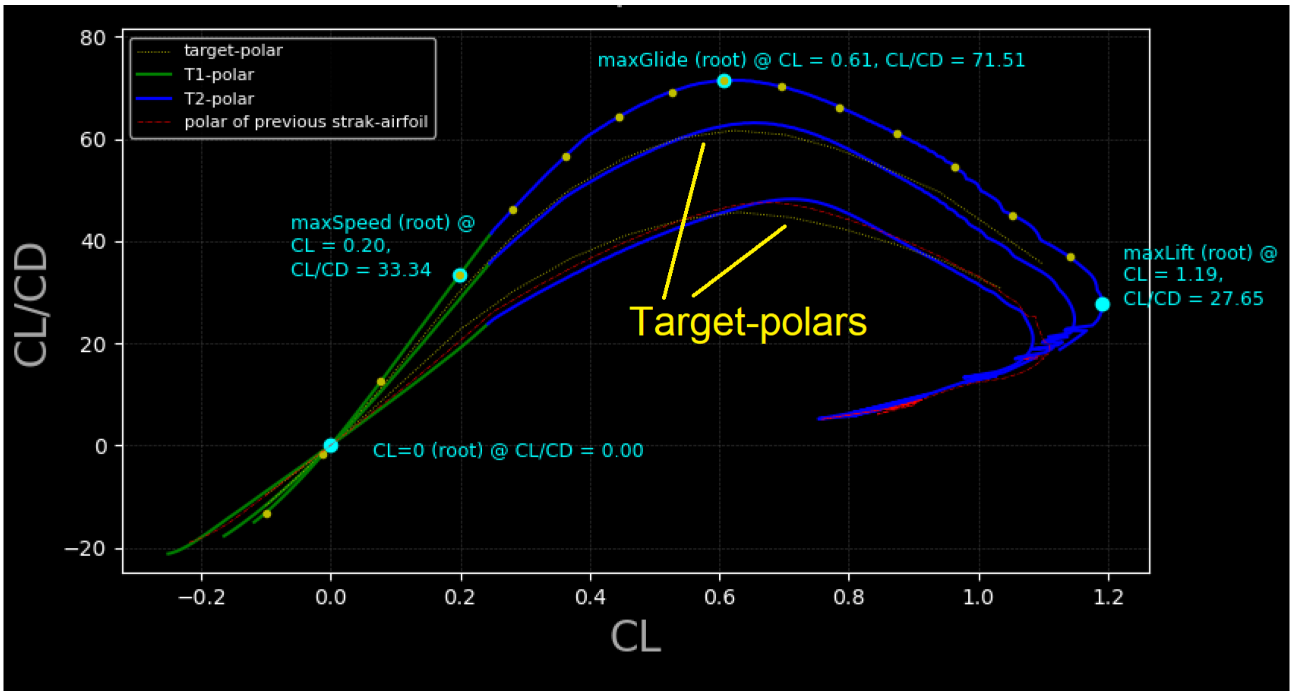
The target-polars mainly consist of discrete Lift/Drag-operating points at a certain Reynolds-number.

By default, 15 discrete points will be used to describe the target-polar and one additional point will be used for optimization of the zero lift angle (which is not visible in the graphs).

The number of points can be changed by the user on demand (for more information see „Strak_Machine_Reference“)

When you look in the main-window of the Strak Machine, you will find the target“-polars as yellow-dotted lines inside two of the graphs (the discrete points being connected by straight lines).





Every target-polar runs through four points, that can be trimmed by the user, so trimming these points will affect the shape of the whole target-polar.

These four points are (sorted by C_L increasing from minimum to maximum):

minC_L:	point of the target-polar with the smallest C_L-coordinate
maxSpeed:	point of the target-polar with the smallest C_D-coordinate
maxGlide:	point of the target-polar with the highest C_L/C_D-ratio
maxLift:	point of the target-polar with the highest C_L-coordinate

Each of these points has a „Gain“-trimming-value.
The „maxGlide“ point also has a „Shift“-trimming-value.

The function of the „Gain“-trimming-values is the same for each of the points.
The Gain-Trimming-value will change C_D of the points mentioned above, keeping the value of C_L unchanged.

The value 0.0 for the „Gain“-trimming-value means, that C_D of the target-polar-point will have the same value as C_D of the root-polar-point (at the Re-number where the airfoil shall be used).

A negative value means, that C_D of the target-polar-point will have a higher value as C_D of the root-polar-point (more drag, deterioration).

A positive value means, that C_D of the target-polar-point will have a lower value as C_D of the root-polar-point (less drag, improvement).

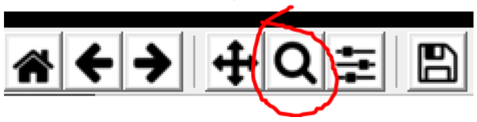
Setting the value to 1 means (not recommended, but for better understanding, how it works and how the standardization is), that the target-polar in this point at the lower Re-number shall have the same (low) drag as the root-polar at its original Re-number.

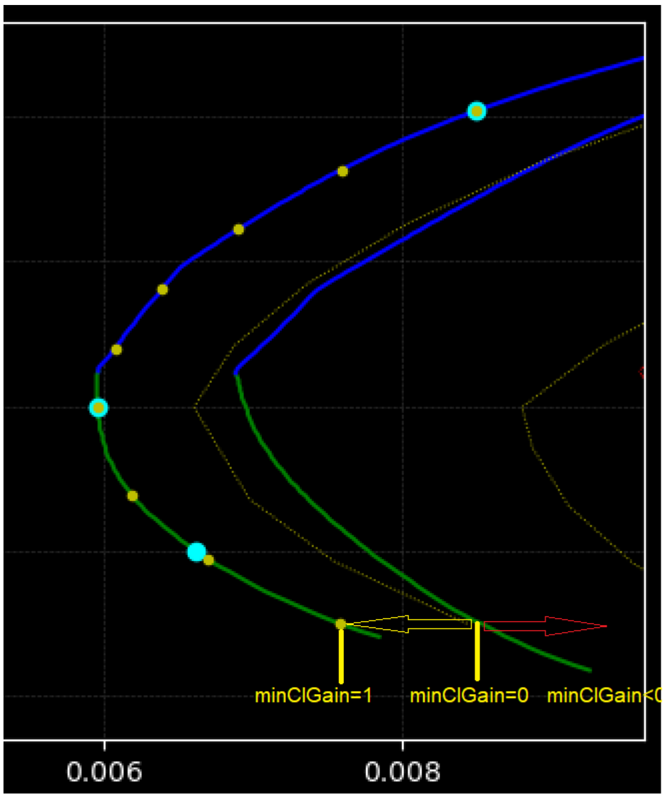
The next passage will show some pictures how the „Gain“-trimming-values work, to get a better understanding.

„minCLGain“

This trim-value will move the most lower point of the target polar to the left or right (if you look at the „C_L over C_D“ - graph).

The picture below shows the graph „C_L over C_D“ zoomed in, using the zoom-button in the Strak Machines toolbar, to allow a more detailed view.





If the point is moved more to the left, this means that especially at lower or even negative C_L-values the drag shall be reduced.

Setting this value to „1“ will mean, that the target-polar will meet the polar of the root-airfoil at this point and below „minCL“ will even have a lower drag as the root-polar.

Thus it is not recommended to use values >1, although it would be possible.

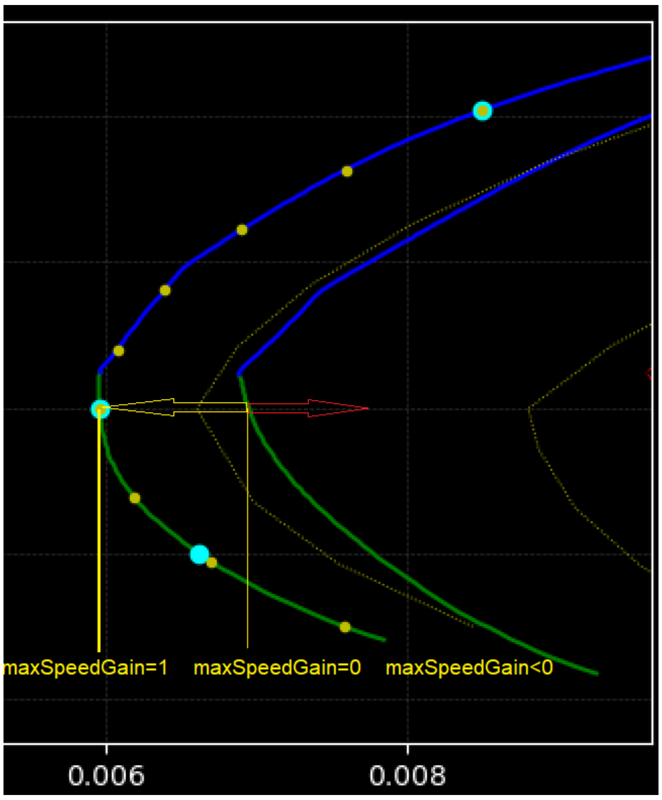
Learning what happens and how to use the Strak Machine:
Try what happens to the geometry of the airfoil, if you move only this point to the left and to the right.
Change the second value in the list „“minClGain“ in „strakdata.txt“ using a text-editor.
Then restart the Strak Machine to make the change become effective.
Look at the graph, how has the target polar been changed?
Then generate an airfoil using the batch-file „make_150k.bat“.
You can have a quick look at the ongoing optimization, if you start „visu_150k.bat“
You can also do this with the other trim-values to learn what happens.

"maxSpeedGain"

This trim-value will move the point of lowest drag of the target-polar to the left or right (if you look at the „C_L over C_D“ – graph).

If the point is moved to the left, this means that lower drag is requested, which will most probably lead to a thinner airfoil to be created by Xoptfoil.

You normally would like to move this point a little bit to the left, using a value between 0 and 0.5, so that the drag of the target polar is lower than the drag of the polar of the root-airfoil at this Re-number, like it is shown in the picture below.



"maxGlideGain"

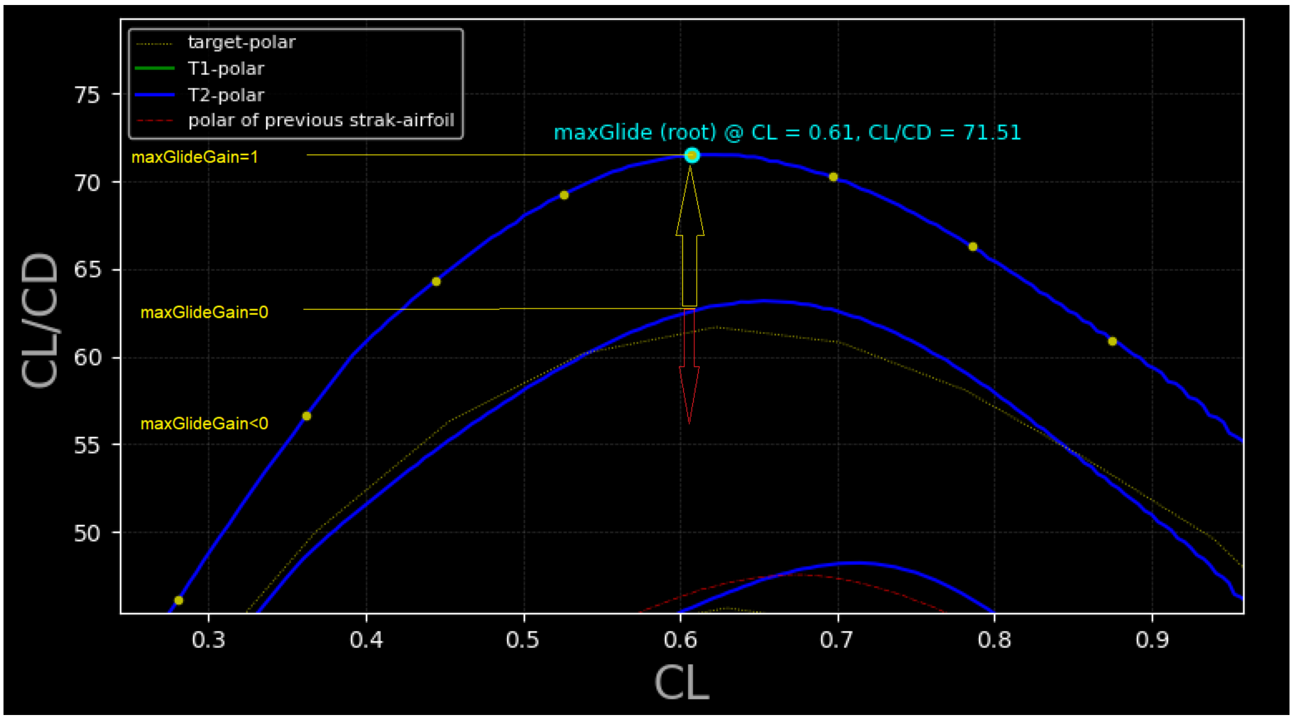
Changing the shape of the polar of an airfoil very often means: to get a desired improvement in a certain area of the polar, something has to be sacrificed in another area of the polar.

If the overall desired improvements are set too high (e.g. improve maxSpeed, also improve maxGlide, improve drag at minCL and improve maxLift), the response of the Xoptfoil-optimization will probably be, that only some of the targets can be achieved and others will be not or get even worse then they have been before.

The polar looses its targeted shape / gets deformed and will not harmonize with the polar of the root-airfoil anymore.

So in the example, that is shown here, it is the C_L/C_D ratio, where something will be sacrificed, to allow the desired improvements in the other areas.

In the picture below you can see how the maxGlide-point of the target-polar will be moved down a little bit by using a negative value for „maxGlideGain“, here it is the value -0.14.

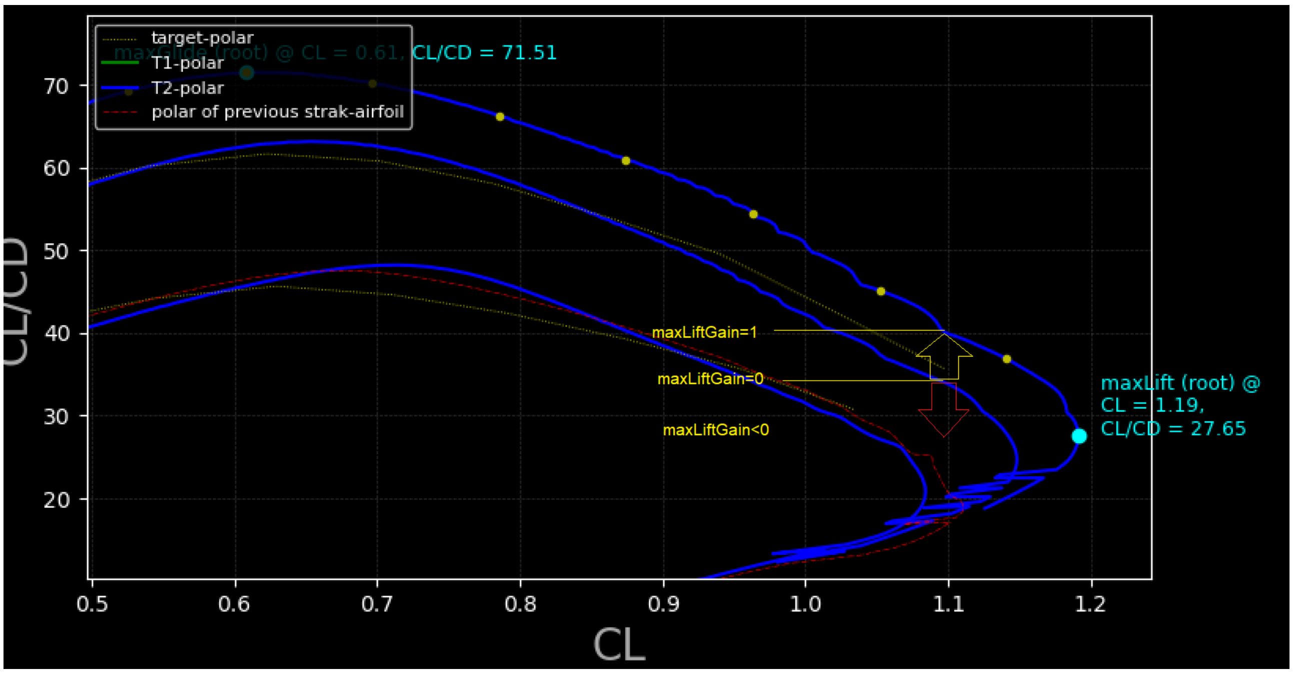


Learning what happens and how to use the Strak Machine:
Try what happens, if you only move the maxGlide-point, keeping all other trim-values untouched.
How will that have an impact on the geometry of the airfoil?
How does camber / thickness change?
How will that eventually have an impact of the maximum Lift that can be generated by the airfoil?

"maxLiftGain"

This trim-value will have an influence on the operating point with the highest Lift of the target polar.

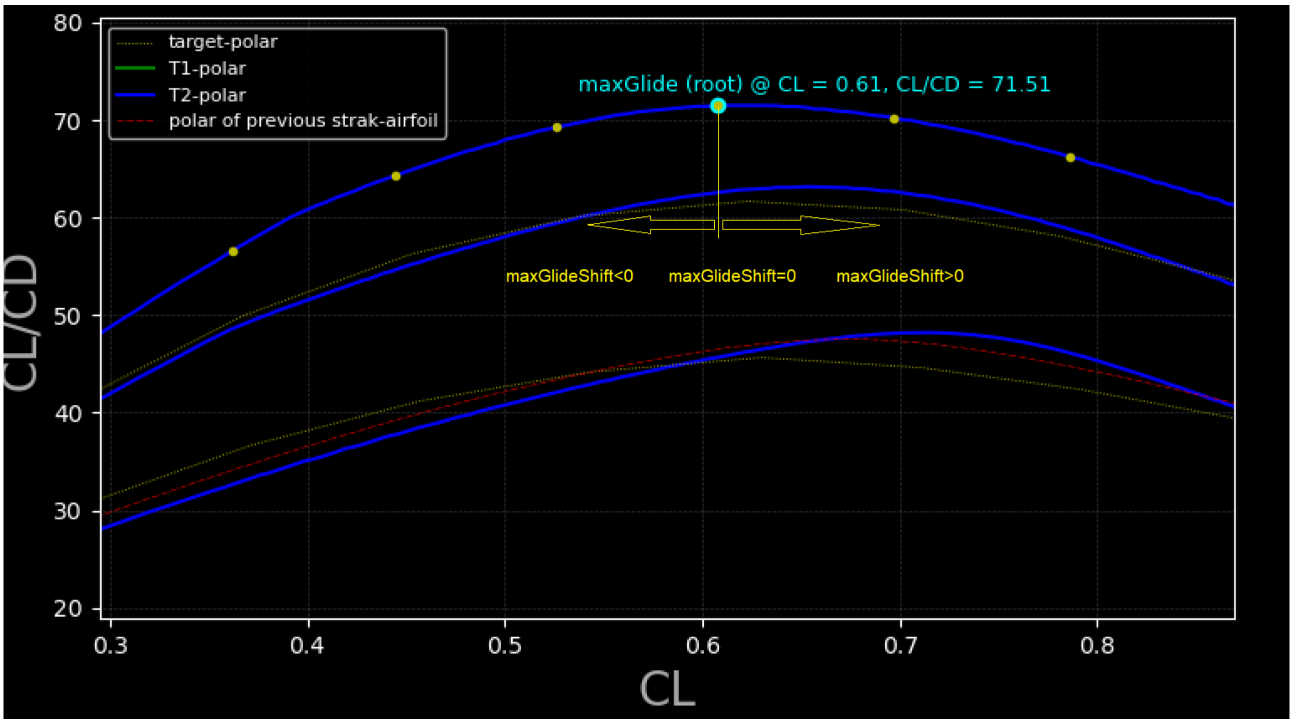
As it is shown in the picture below, this point can be shifted up and down using the trim-value "maxLiftGain".
As with the other previously mentioned trim values, it is not recommended to use values > 1.



"maxGlideShift"

The value of „maxGlideShift“ specifies a C_L-difference for shifting the maxGlide-point (change C_L of the maxGlide point).

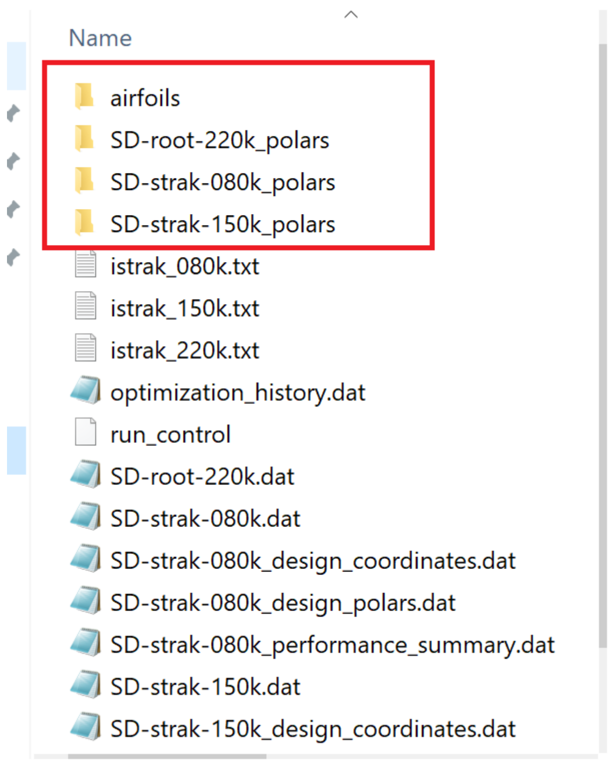
This trim-value will often change the x position of max camber of the generated airfoil and can be used to take influence on it.



5 Looking at the example data

The Strak Machine comes ready prepared with an example „SD-strak“.

You will find all of the prepared results /airfoils / polars inside the „build“ folder.

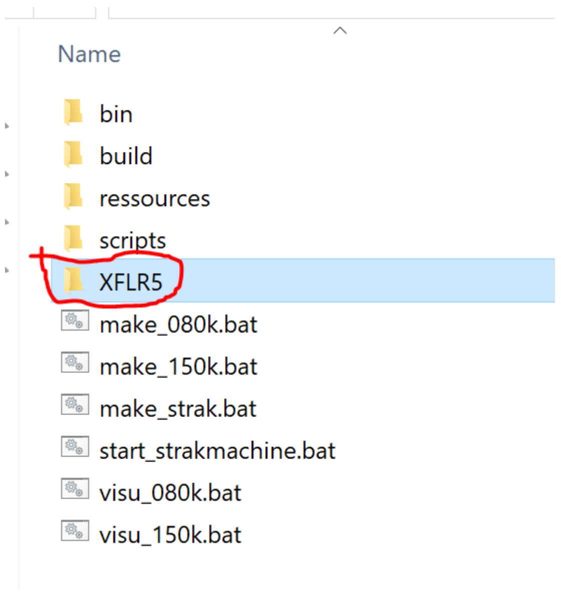

Name

airfoils
SD-root-220k_polars
SD-strak-080k_polars
SD-strak-150k_polars
istrak_080k.txt
istrak_150k.txt
istrak_220k.txt
optimization_history.dat
run_control
SD-root-220k.dat
SD-strak-080k.dat
SD-strak-080k_design_coordinates.dat
SD-strak-080k_design_polars.dat
SD-strak-080k_performance_summary.dat
SD-strak-150k.dat
SD-strak-150k_design_coordinates.dat

Inside the „build“-folder there is an „airfoil“-folder, where all of the generated airfoils will be stored. Inside the „build“-folder there are also further „...polars…“ folders, that contain polars of the generated airfoils.

Before you start your own strak-generation, you want to look at the ready generated airfoils and polars as a reference.

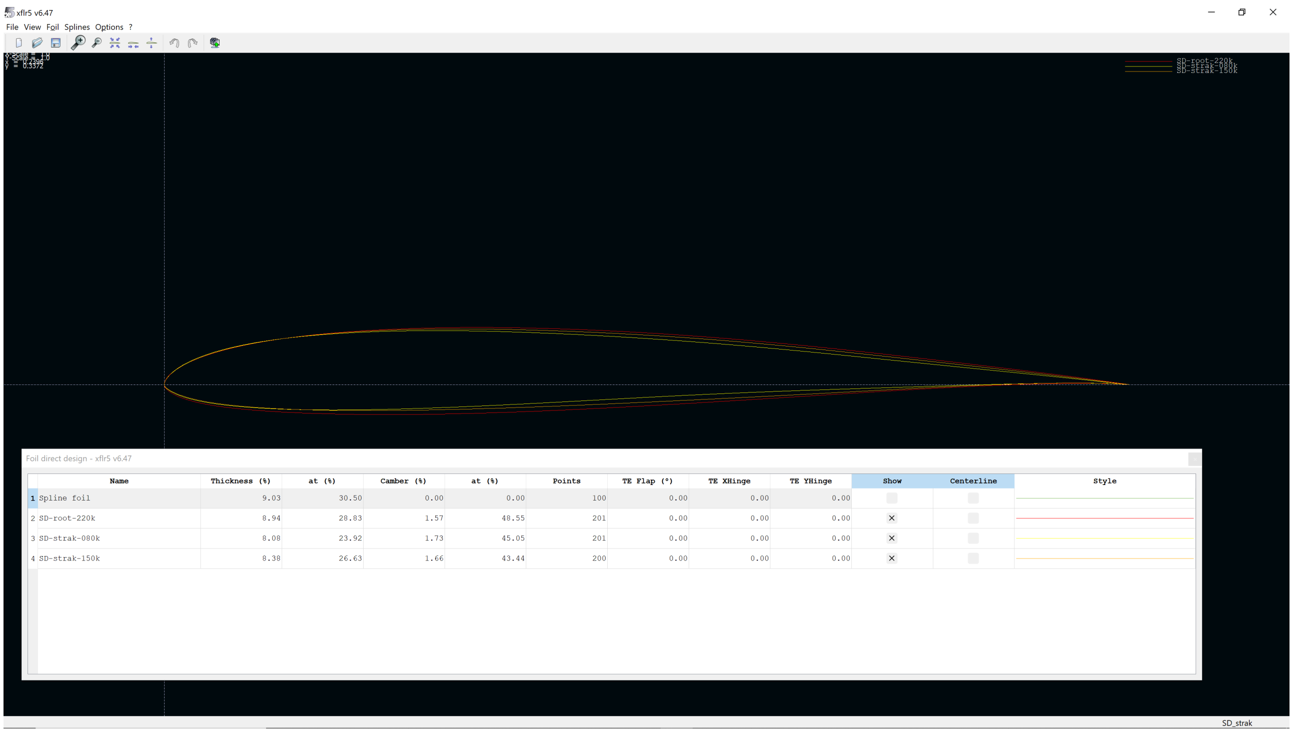
You can either import them in a tool like XFLR5 on your own, or you can open the prepared XFLR5-project-file that can be found in the „XFLR5“-folder (if you have already an installed version of XFRL5).


Name

bin
build
ressources
scripts
XFLR5
make_080k.bat
make_150k.bat
make_strak.bat
start_strakmachine.bat
visu_080k.bat
visu_150k.bat

After opening the prepared XFLR5-Project, you should see the root-airfoil and the strak-airfoils, that have been created with the example-configuration.

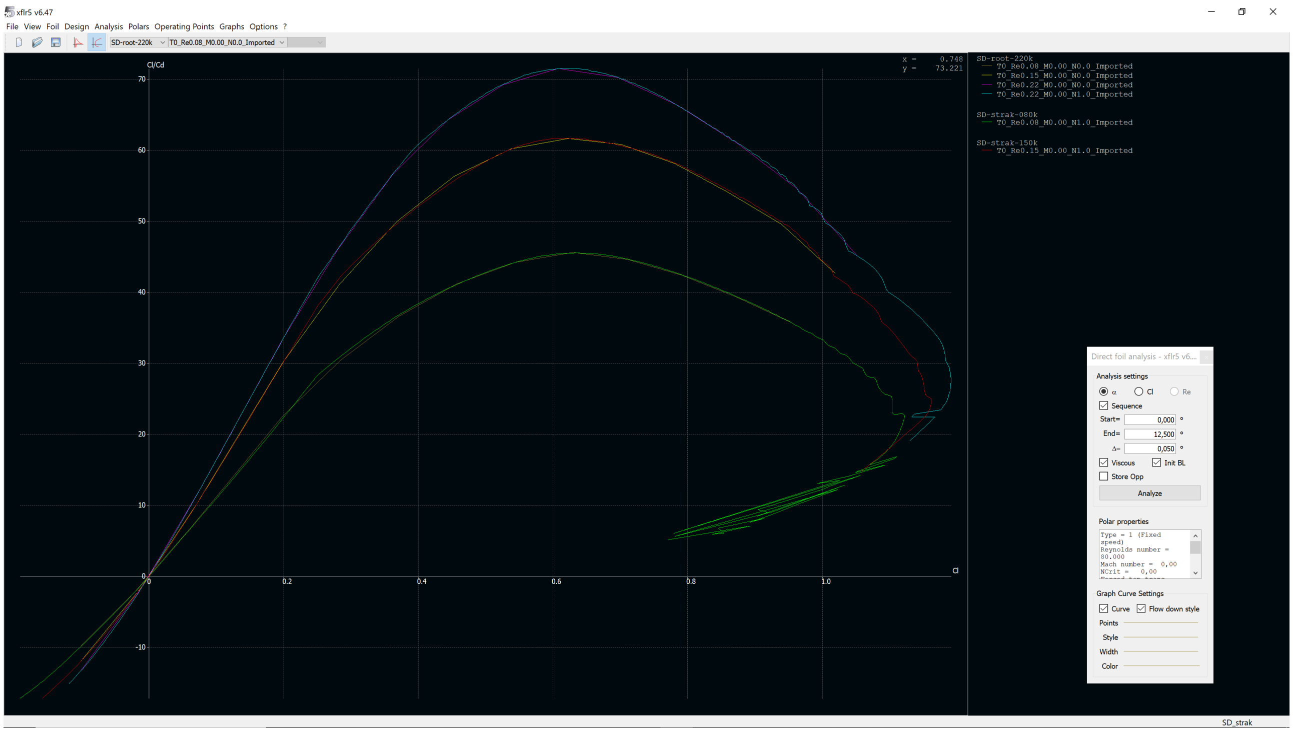
If the window does not show at once, you have to use the menu-entry „File→Direct foil Design“.



Foil direct design - xflr5 v6.47

	Name	Thickness (%)	at (%)	Camber (%)	at (%)	Points	TE Flap (°)	TE XHinge	TE YHinge	Show	Centerline	Style
1	Spline foil	9.03	30.50	0.00	0.00	100	0.00	0.00	0.00			
2	SD-root-220k	8.94	28.83	1.57	48.55	201	0.00	0.00	0.00	✕		
3	SD-strak-080k	8.08	23.92	1.73	45.05	201	0.00	0.00	0.00	✕		
4	SD-strak-150k	8.38	26.63	1.66	43.44	200	0.00	0.00	0.00	✕		

The polars can be shown using the menu-entry „File→Xfoil Direct Analysis“


xflr5 v6.47
File View Foil Design Analysis Polars Operating Points Graphs Options ?
SD-root-220k
T0_Re0.08_M0.00_N0.0_Imported
Cl/Cd
70
60
50
40
30
20
10
0
-10
0
0.2
0.4
0.6
0.8
1.0
Cl
x = 0.748
y = 73.221
SD_strak

SD-root-220k
- T0_Re0.08_M0.00_N0.0_Imported
- T0_Re0.15_M0.00_N0.0_Imported
- T0_Re0.22_M0.00_N0.0_Imported
- T0_Re0.22_M0.00_N1.0_Imported

SD-strak-080k
- T0_Re0.08_M0.00_N1.0_Imported

SD-strak-150k
- T0_Re0.15_M0.00_N1.0_Imported

Direct foil analysis - xflr5 v6....

Analysis settings
[✓ α] Cl Re
[✓] Sequence
Start= 0,000 °
End= 12,500 °
Δ= 0,050 °
[✓] Viscous [✓] Init BL
[ ] Store Opp
Analyze

Polar properties
```
Type = 1 (Fixed
speed)
Reynolds number =
80.000
Mach number =  0,00
NCrit =   0,00
```

Graph Curve Settings
[✓] Curve [✓] Flow down style
Points
Style
Width
Color

The usage of the number-keys „1“ .. „5“ allow fast switching between the different polar-graphs.

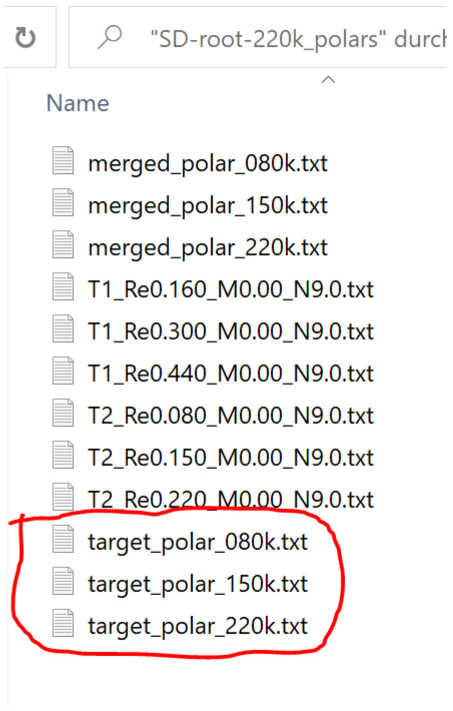
In the picture above you will see the calculated and merged type 2 / type 1-polars of the three airfoils included in the project.

The somewhat „square“ looking polars are the „target“-polars that have been generated by the Strak Machine (remember: only a few discrete points)

You can compare, how much the polars of the generated airfoils differ from the target-polars (in the best case there should be only small differences).

A few more words concerning the target-polars:

All target-polars can be found in the folder „SD-root-220k_polars“ (or express it more „generic“: inside the polar folder of the root-airfoil)


"SD-root-220k_polars" durch

Name

merged_polar_080k.txt
merged_polar_150k.txt
merged_polar_220k.txt
T1_Re0.160_M0.00_N9.0.txt
T1_Re0.300_M0.00_N9.0.txt
T1_Re0.440_M0.00_N9.0.txt
T2_Re0.080_M0.00_N9.0.txt
T2_Re0.150_M0.00_N9.0.txt
T2_Re0.220_M0.00_N9.0.txt
target_polar_080k.txt
target_polar_150k.txt
target_polar_220k.txt

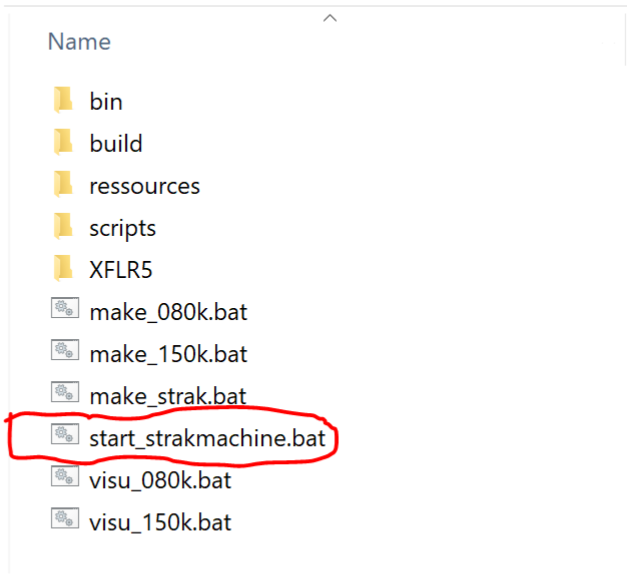
That means, that they will be associated with the root-airfoil.
The reason is, that XFLR5 will only display polars of an airfoil, if the airfoil was already inserted into the project.

In case of having only a root airfoil and not having created the strak-airfoils, you can already import the target-polars into XFRL5 to take a better look at them than you can do with the Strak Machine.

6 Generating your first strak based on the example

You can first apply your desired changes to „strakdata.txt“, e.g. change the „maxSpeedGain“-value of the first strak-airfoil for testing purposes.

Then start the Strak Machine by running „start_strakmachine.bat“.



Name

- bin
- build
- ressources
- scripts
- XFLR5
- make_080k.bat
- make_150k.bat
- make_strak.bat
- start_strakmachine.bat
- visu_080k.bat
- visu_150k.bat

When you see the main window, the Strak Machine will already have written all files to your hard-drive, that are necessary for the strak-generation.

The batchfiles „make_strak.bat“, make_150k.bat“, „make_80k.bat“, visu_150k.bat“, „visu_80k.bat“, will either be created or overwritten, if they already existed.

Also the build-folder will be created, if it does not exist and initially filled.

You can start the strak-generation either by running the batchfile „make_strak.bat“ (to generate the whole strak) or by starting „make_150k.bat“ (to generate the first strak-airfoil).

The results inside the „build“-folder now will be overwritten, so if you want to keep all airfoils / polars as a reference, you should make a backup first.

If you choose the „step-by-step“ method to generate your strak, it is normally necessary to create the airfoils „from root to tip“, so call “make_150k.bat“ first, then call „make_80k.bat“.

But as the airfoil „SD-strak-150k.dat“ has been already created in the example, you can also directly call „make_80k.bat“.

The ongoing optimization can be monitored by running the batchfile „visu_150k.bat“ or „visu_80k.bat“, depending on which airfoil is being created.

This will take some time, depending on the computing power of your PC (as a reference: the generation of an airfoil with „Hicks-Henne“ will take approximately 1 hour on an AMD Ryzen 3900 12 core@4GHz. Multicore will help, as Xoptfoil will execute parallel threads).

After the optimization has been finished, you can import the generated airfoil e.g. „SD-strak-150k.dat“ into XFLR5, that can be found in the „build\airfoils“-folder.

When you have imported the airfoil, you can afterwards import the polars that can be found in „build\SD-strak-150k_polars“.

7 Changing the root-airfoil, changing Re-numbers

If you would like to use another root-airfoil as the SD-airfoil, you will have to put the .dat-file into the ressources-folder.

Then you have to open „strakdata.txt“ with a text-editor and enter the name of the root-airfoil there ("seedFoilName").

You can also change Re-numbers in strakdata.txt according to your needs or add Re-numbers to create further airfoils.

If you want to create a different number of strak-airfoils / more airfoils than have been used in the example, you will not only have to add the Re-numbers in the „reynolds“-list, but also have to add the corresponding „trim-values“ (like „maxGlideGain“ etc.) for the target polars.

The Re-numbers in the „reynolds“-list must be sorted descending.

If the Strak Machine does not start properly after you have changed „strakdata.txt“, it will probably be a problem that the json-parser has detected.

This will occur, if you have forgotten a comma or if another „formal“-problem is there inside the file „strakdata.txt“. In this case please check, what you have changed last to find the error.

It is recommended that you delete the build-folder, before you start the Strak Machine with the new airfoil-settings, to keep an overview what dat belongs to the new strak and not to mix with data from the previous project.

After having applied all of your changes to strakdata.txt, close the file and start the Strak Machine.

Now it will take some time for the polars to be calculated and the main window to appear, because the Strak Machine will not find previously calculated polars for the new airfoil / the new Re-numbers.

The polar calculation will also been carried out automatically, if you only change „maxReFactor“ (all type 1 polars have to be calculated).

After changing airfoil and / or Re-numbers you should also adapt the trim-values of the polars in „strakdata.txt“.

Known problems with polar calculation / fixes:
For a proper analysis of the root-airfoil it is necessary to have the type 1 polars calculated with a minimum C_L of -0,1 (this is the default-value, the minimum C_L may also be changed by a parameter in strakdata.txt.)

The polar-calculation will be controlled by the file „iPolars_T1.txt“ that can be found in the „ressources“ folder.

The content should look like this:

```
! Helper inputs for polar generation

&polar_generation
  generate_polars = .true.
  type_of_polar   = 1
  op_mode         = 'spec-al'
  op_point_range  = -3, 12, 0.05
```

```
/

&xfoil_run_options
  reinitialize = .true.
/
```

Depending on the root-airfoil, it may be necessary to change the yellow-marked value, which is the minimum alpha, from -3 to -4 or even -5 degrees, to get C_L = -0.1 in the polar.

You will have to test to change the value, if the Strak Machine has been aborted with an error-message after importing the polars.

You can also look yourself in the type 1-polar-file of the root-airfoil with a text-editor and check whether the polar starts below C_L = -0.1.

Th e polr can be found in the ...polars-folder in „build\“ .